TaskLink™ for Windows® User Manual

UniSystem Programming System

Data I/O

XXX-XXXX-XXX

March 2000 xxx-xxxx-xx

Data I/O has made every attempt to ensure that the information in this document is accurate and complete. Data I/O assumes no liability for errors, or for any incidental, consequential, indirect, or special damages, including, without limitation, loss of use, loss or alteration of data, delays, or lost profits or savings, arising from the use of this document or the product which it accompanies.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without written permission from Data I/O.

Data I/O Corporation
10525 Willows Road N.E., P.O. Box 97046
Redmond, Washington 98073-9746 USA
425-881-6444

www.dataio.com

Acknowledgments:

Data I/O is a registered trademark and TaskLink is a trademark of Data I/O Corporation.

Data I/O Corporation acknowledges the trademarks of other organizations for their respective products or services mentioned in this document.

©1999/2000 Data I/O Corporation
All rights reserved

Table of Contents

TASKLINK FOR WINDOWS: OVERVIEW 1

INTRODUCTION 1
System Control *1*
Tasks *1*
Data Transfer *2*
Displays and Controls *2*
Additional Features *2*
SYSTEM REQUIREMENTS 2
Computer System *2*
Programmers *2*
TASK LINK USER INTERFACE 2
TaskLink Main Screen *3*
System Status 3
Job Info 3
Last Job Summary 3
Programmer Status 3
TaskLink Toolbar 4
ABOUT TASKS AND KITS 5
Tasks *5*
Kits *5*

TASKLINK AND UNISYSTEM PROGRAMMERS 6

TASKLINK: UNISYSTEM MENUS 7

SETUP MENU 7
Select Device *7*
Custom Device File 8
Programmer Properties *9*
Process 10
General 12
Memory 14
Logic 16
Sector 17
Special 19
Serialize 20
Restore Defaults *21*
Session Data Logging *21*
Log File 22
Log File Format 22
Exit *23*
DATA MENU 24
Load RAM from PC/Network File *24*
Translation Format 25
Load RAM from Programmer File *26*
Load RAM from Master *26*
Load RAM from Host *26*
Host Command 27
Save RAM to PC/Network File *27*
Translation Format 28
Save RAM to Programmer File *28*
Copy Data from PC File to Programmer File *28*
Copy Data from Programmer File to PC File *28*
Copy Data from Master Device to Programmer File *28*
View/Edit RAM *29*

Fill RAM *29*
Fill Fuse Map *29*
Delete Programmer File *30*
PROCESS MENU 31
Process Devices *32*
Select Process *33*
Blank Check Only *33*
Illegal Bit Check Only *33*
Program/Verify Only *33*
Verify Only *34*
TASK MENU 35
Task Database Device Check *35*
Run Batch File *35*
Run Last Task/Kit (listed by filename) *36*
TOOLS MENU 37
Programmer Interface *37*
VT100 on Programmer Port/Host Port *37*
Device Database Update *37*
SYSTEM MENU 38
Select Programming System *38*
System Security *39*
Username 39
Password 39
Printer Setup *39*
Set Message Editor *39*
Create Messages 40
Simulation Mode *40*
Options *40*
General 41
Programmer 41
Port Settings 42
Logging 42
HELP MENU 43

TASKLINK: UNISYSTEM OPERATIONS 44

EDIT A TASK 44
Task *45*
Process *47*
Data *48*
Memory *49*
Logic *50*
Sector *51*
Special *52*
Serialize *53*
General *54*
EDIT A KIT 54
USE THE TASK/KIT MANAGER 55
Task File *55*
Run *55*
Add *55*
Delete *55*
Edit *55*
Duplicate *55*
Import *55*
Export *55*
Search *55*
View *55*

ADD A TASK TO A KIT 56
To add a Task to a Kit: *56*
ADD NEW USERS TO TASKLINK 57
RUN A KIT OR TASK 58
USE SERIALIZ.EXE 59
WRITE A SERIALIZATION PROGRAM 61
Communication with an ESP *61*
Command Line Arguments *61*
The Output File SERIAL.DAT *62*
T01: Current serial number(s) 62
T02: Next serial number 63
T03: Translation format number 63
T04: "Beginning of serial number data" marker 63
Example SERIAL.DAT file *63*
T00: "Pure-set" gang serialization flag (TaskLink 1.1A) 63
T05: Fatal-error message 64
T07: Hexadecimal address of serial number in device 64
T08: Decimal field length of serial number in device 64
T09: RAM checksum 64
T11: Warning message 64
Example SERIAL.DAT file *64*

DATA I/O: SERVICE AND SUPPORT 65

CUSTOMER SUPPORT OFFICES 65
United States *65*
Canada *65*
Japan *65*
Germany *65*
Other Countries *65*
END USER REGISTRATION 65
SOFTWARE UPDATE SERVICE 66
UniFamily Programming System *66*
ProLINE-RoadRunner Programming System *66*
Sprint Family Programming System *66*
TECHNICAL ASSISTANCE 67
Calling *67*
Electronic Mail *67*
Data I/O Web Site *67*
WARRANTY INFORMATION 67
COMMAND LINE PARAMETERS 67
KEY TERMS 68
Administrator Mode *68*
Operator Mode *68*
Data Source *68*
Checksum *68*
Process *68*
Notes *68*
Counts *68*
Times *68*
%Yield *68*

TASKLINK FOR WINDOWS: OVERVIEW

***Note**: The "TaskLink for Windows: Overview" section contains general information about TaskLink for Windows 4.0 and is not specific to the UniFamily Programming System.*

Introduction

TaskLink™ for Windows 4.0 automates and controls Data I/O's device programming systems, enhancing the quality and efficiency of your device programming operations. TaskLink provides a consistent, simple to use interface for a wide mix of applications. With TaskLink, you can standardize operations for improved productivity, greater flexibility, and reduced costs.

TaskLink simplifies setup and execution of device programming to ensure every batch is error-free. For example, you can easily capture the system configuration and processing steps needed to program each part number in your inventory by creating tasks. Tasks specify the device type, data file, processing, and other settings. Because a user-defined name and description reference them, tasks are easily selected by the operator for fast, automated setup and control. In fact, TaskLink's task features, combined with high-speed download capabilities, can reduce errors and make setup nearly ten times faster.

Sophisticated tracking features also help to maximize quality. TaskLink automatically tracks the number of devices programmed, so every batch contains the right quantity. In addition, it logs and displays detailed statistics about each programming session, so you can analyze trends and zero in on how to improve quality in your production process.

For added safety and efficiency, TaskLink features two user modes.

- Administrator Mode provides full access to the complete range of TaskLink features, including task creation and editing.
- Operator Mode only allows execution of predefined, automated tasks, ensuring maximum efficiency while safeguarding against errors.

TaskLink also streamlines operation by storing all programming data files on a hard disk, a network file server, or a host system, eliminating the need for cumbersome master devices.

System Control

- Simple to use interface provides integrated control of all system components, including the device programming system and data source.
- Supports a wide variety of Data I/O device programming systems.
- Flexible programming control lets you choose any combination of blank, continuity, and illegal-bit checks, array and security fuse programming, and fuse and functional verification operations.
- Flexible memory programming control options for fill, set-size, word-width, and address offset.

Tasks

- Forms-based editor prompts you for task parameters by listing options.
- A broad range of parameters is available, including device type, data file, I/O, processing, verify passes, reject count, and memory-device/logic-device parameters.
- Task names and descriptions can be customized.
- Multiple task databases can be created, limited only by available disk space.
- Enhanced automation means that tasks can be linked together to automatically reconfigure the system between batches of parts in the same kit.
- Extensive task-management facilities, such as edit, delete, copy and search, maximize power and flexibility.
- Detailed programming session logs–including error conditions like continuity fail, non-blank fail, program fail, fuse-verify fail, and functional-test fail–help you analyze programming trends and better manage quality.

- Automatic tracking of the number of devices successfully programmed helps ensure accurate and consistent job completion.

Data Transfer

- Programming data can come from a disk file, network file server, remote host file, or master device.
- High-speed data transfers via serial 115K baud allow fast downloading and uploading of large files.
- Support for all popular I/O protocols, including binary, Motorola®, and Intel®.
- Built-in terminal emulator provides instant access to a host or device programming system.

Displays and Controls

- Windows® interface features simple pull-down menus, dialog boxes, and your choice of mouse and/or keyboard control.
- Context-sensitive, on-line help provides immediate assistance at the touch of a button.
- Two user modes, Administrator and Operator, tailor the interface and control options to the user's skill level, ensuring flexible, fast, and safe operation.
- Device selection by manufacturer name and part number, or by a wild card search (? or *), streamlines the operation; device names can also be customized with your own part numbers.

Additional Features

- External program interface lets you generate Ethernet addresses and other unique device data for automated production of devices with serial numbers.
- TaskLink's device selection menus are updated from the programmer's device list whenever the programmer is updated, so TaskLink always stays current with your device programming systems.

System Requirements

Computer System

- Microsoft® Windows 95/98/NT
- Mouse (Bus or Serial)
- VGA Monitor, 640x480 minimum resolution
- CD-ROM Drive
- Data Cable (shipped with programmer)

Programmers

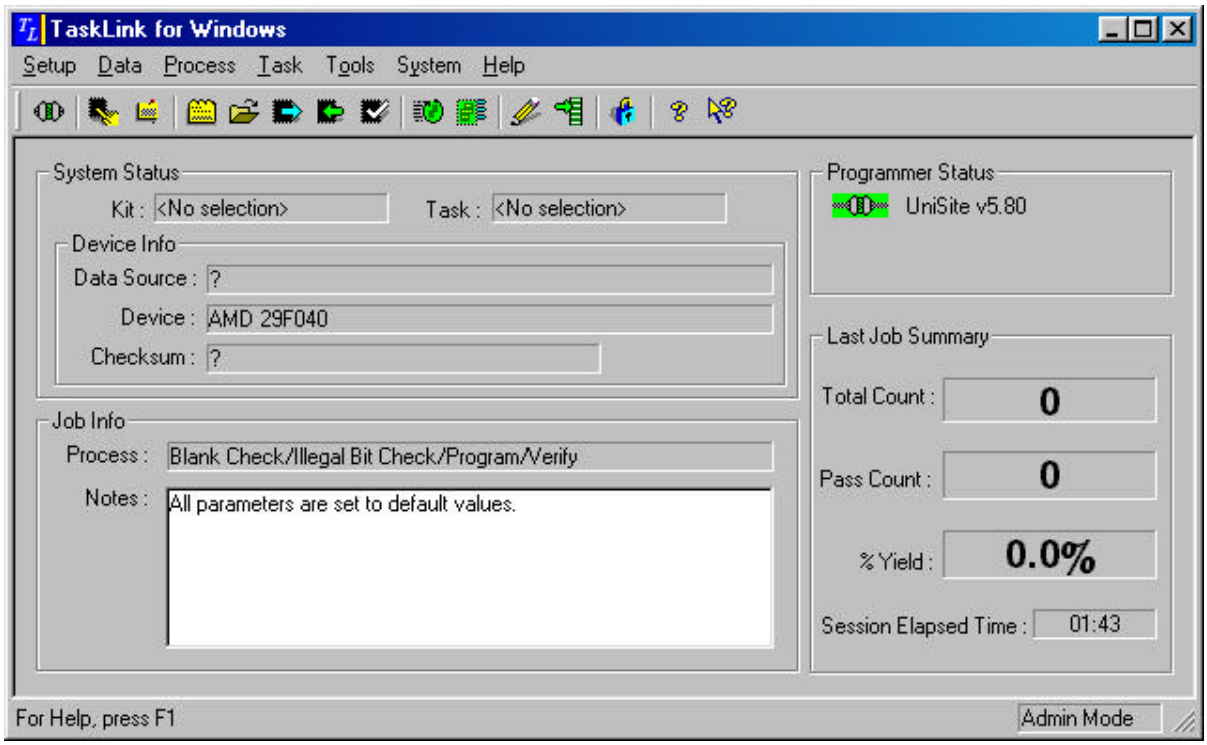
- UniSite™ Universal Programmer with PinSite™ Programming Module
- AutoSite™ Production Programmer (stand alone mode)
- 2900 Programming System
- 3900/3980 Programming System
- ProLINE-RoadRunner System
- Sprint Family Programming System

TaskLink User Interface

The TaskLink User Interface refers to the systematic organization of screens and menus that you see when using the TaskLink application. TaskLink uses many menus and many different screens. Depending on the programmers and handlers that are configured, many variations exist in the TaskLink screens.

TaskLink Main Screen

One screen that remains fairly constant is the TaskLink main screen. The main screen is the first screen you see when starting TaskLink. An example of the TaskLink main screen is shown below.



The Main Screen displays the following groups of information:

SYSTEM STATUS

Displays the name of the current Task or Kit as well as device information including the file name and location of the Data Source, make and model of the device, and any Checksum value.

JOB INFO

Displays the current process and any Notes associated with the process.

LAST JOB SUMMARY

Displays statistics from the last job including Counts, %Yield, and Times.

PROGRAMMER STATUS

Displays the connection status to the programmer and which programmer is used.

- This symbol indicates that contact has been established with the programmer.

- This symbol indicates that there is no contact with the programmer.

TASKLINK TOOLBAR

The TaskLink Toolbar contains shortcuts to many of the most often-used menu items. The toolbar can change depending on which programmer you have configured. An example of the toolbar is shown below.

Each icon in the toolbar corresponds to a TaskLink menu item. To quickly select the menu item, left-click the toolbar icon. The menu items that match the toolbar icons are listed below:

- Click this button to establish contact with your programmer.

- Click this button to select a device from the list of supported devices.

- Click this button to view the footnotes for the selected device.

- Click this button to set programmer properties.

- Click this button to load the programmer RAM with data from a PC file.

- Click this button to load the programmer RAM with data from a master device.

- Click this button to perform selected processes on a device.

- Click this button to verify a device.

- Click this button to run a Task or Kit.

- Click this button to manage Tasks and Kits.

- Click this button to view or edit programmer RAM data.

- Click this button to fill a programmer RAM block with specific data.

- Click this button to set the TaskLink security level and add users.

- Click this button view online Help.

- Click this button to get context-sensitive Help.

About Tasks and Kits

TaskLink uses Tasks and Kits to automate complicated programming jobs. A Task is the TaskLink term that describes a single programming job. A Kit is the TaskLink term that describes a sequence of programming jobs. Tasks and Kits are the two types of data records found in a Task/Kit Database file (*.TSK).
A Task/Kit database (*.TSK) contains records which can be only Tasks, only Kits, or a combination of both. Options to add, remove, and edit Tasks and Kits are found in the Task/Kit Manager screen.

Tasks

A Task record contains all the information necessary for programming and testing one or more devices with a particular data pattern. Examples of the information contained in a Task are:

Device type: Specific device types or a list for operator selection.

Data source: PC disk file, master device, or download from a remote host.

Process: Sequence of operations you want to perform on the device, such as illegal bit check, blank check, program, and verify.

Programming: Programming/testing options and available parameters on Data I/O programmers.

Serialization: Used to program each device with unique information.

Kits

A Kit record consists, for the most part, of a list of Tasks. When listed in a Kit, a group of Tasks can be referred to by a 25-character name, and a 40-character description.

Tasks can be organized into Kits to automate the production of a series of programmable devices, such as all of the programmable devices found on a printed circuit board assembly. All of the Tasks in a Kit must come from the same Task/Kit Database file.

When you run a Kit, you can either run all the Tasks in sequence or run only one of the Tasks. Kits can thus be used for two purposes:

- Minimize the chance of operator error by automatically running the correct Tasks to produce a set of programmed devices.
- Provide a method of organizing a group of Tasks.

Another feature of Kits is the ability to specify a numbered message with each Task. When the Kit is run, the messages are displayed for the operator.

TASKLINK AND UNISYSTEM PROGRAMMERS

The Data I/O UniSystem family of programmers includes the 2900 family, the 3900/3980 family, the AutoSite, and the UniSite.

TaskLink™ for Windows® automates and controls the UniSystem family of programmers, enhancing the quality and efficiency of your device programming operations. With TaskLink's simple to use interface to control your UniSystem programmer, you can standardize operations for improved productivity, greater flexibility, and reduced costs.

TaskLink simplifies setup and execution of device programming to ensure every batch is error-free. For example, you can easily capture the system configuration and processing steps needed to program each part number in your inventory by creating Tasks and Kits. Tasks specify the device type, data file, processing, and other settings. Tasks can be organized into Kits to automate the production of a series of programmable devices, such as all of the programmable devices found on a printed circuit board assembly. Because a user-defined name and description reference them, operators can easily select Tasks and Kits for fast, automated setup and control. In fact, TaskLink features, combined with high-speed download capabilities, can reduce errors and make setup nearly ten times faster.

TaskLink for Windows User Manual: UniSystem Programming System is organized to assist you in understanding how to make the best use of TaskLink software. The "TaskLink: UniSystem Menus" section describes the features of each of the seven main menus:

- **Setup Menu** (on page 7)
- **Data Menu** (on page 24)
- **Process Menu** (on page 31)
- **Task Menu** (on page 35)
- **Tools Menu** (on page 37)
- **System Menu** (on page 38)
- **Help Menu** (on page 43)

The "TaskLink: UniSystem Operations" section (on page 44) describes the most frequently used programming and testing operations, including selecting devices, setting programming system options, editing Tasks, running Tasks, exiting TaskLink, and many others.

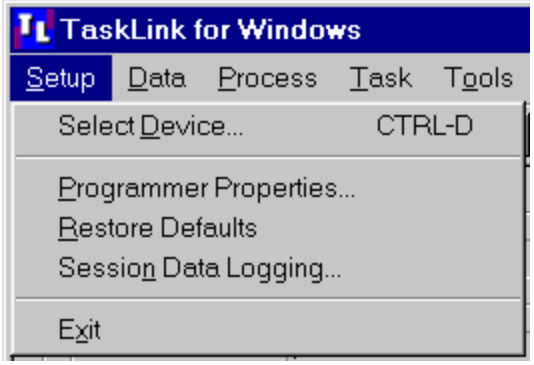
General information about TaskLink for Windows is found in "TaskLink for Windows: Overview" (on page 1), and information about service is found in "Data I/O: Service and Support" (on page 65).

Congratulations on your choice of TaskLink for Windows and Data I/O's UniSystem family of programmers.

TASKLINK: UNISYSTEM MENUS

Setup Menu

The TaskLink Setup Menu is shown below. This menu is used to select device and programmer parameters, to enable TaskLink's data logging features, and to exit from the TaskLink program.

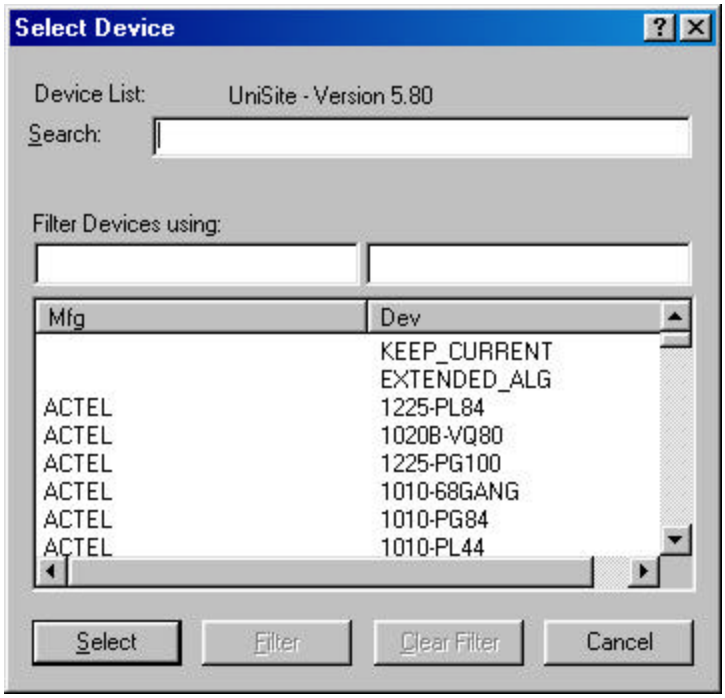


The options on the Setup Menu include:

- Select Device
- Programmer Properties
- Restore Defaults
- Session Data Logging
- Exit

Select Device

The TaskLink for UniSite Select Device screen is shown below.



Use the Select Device screen to specify a target device. You can enter the manufacturer and device name or select the manufacturer and device name from lists. You can also use the filter feature to limit the device display to a small subset of the supported devices. Then use the clear filter button to see all devices again.

Filters
To use filters, enter a standard wildcard value into the Manufacturer and/or Device fields and press the filter button. For example:
Entering: Manufacturer: * and Device: 29?40
Returns a list of all manufacturers and all devices beginning with “29, followed by any character, followed by “040.

CUSTOM DEVICE FILE

Use the Custom Device File feature to create a cross-reference between your own names for programmable devices and the names used by a particular Data I/O programmer.
TaskLink stores device information for each programmer in a separate disk file. The file has a name unique to the programmer and a .dev extension. To use the Custom Device File feature, you must create a file of the same name in the same directory, but with a .cus extension. This file contains the cross-referencing information. A different file is required for each programmer type used.
Once you have created a file, you must run the Device Database Update option from the TaskLink Tools Menu. After running the update, your new device names will appear in the device selection list on the Select Device screen.

Custom Device Filenames
AutoSite: AUTOSITE.CUS
UniSite or UniSite/SetSite: UNISITE.CUS
2900 Programming System: 2900.CUS
3900/3980 Programming System:3900.CUS

Custom Device File Format
A typical custom device file line looks like this:

YourName :
OurName1
OurName2
OurName3

where "YourName :" is your name for the device and "OurName" is the name of the device as it appears in the Select Device menu. You can assign one device or a list of devices to YourName. If you assign several devices to YourName, TaskLink prompts you to choose one when processing devices. If just one OurName is to be assigned to YourName, you can enter a line like this:

YourName : OurName

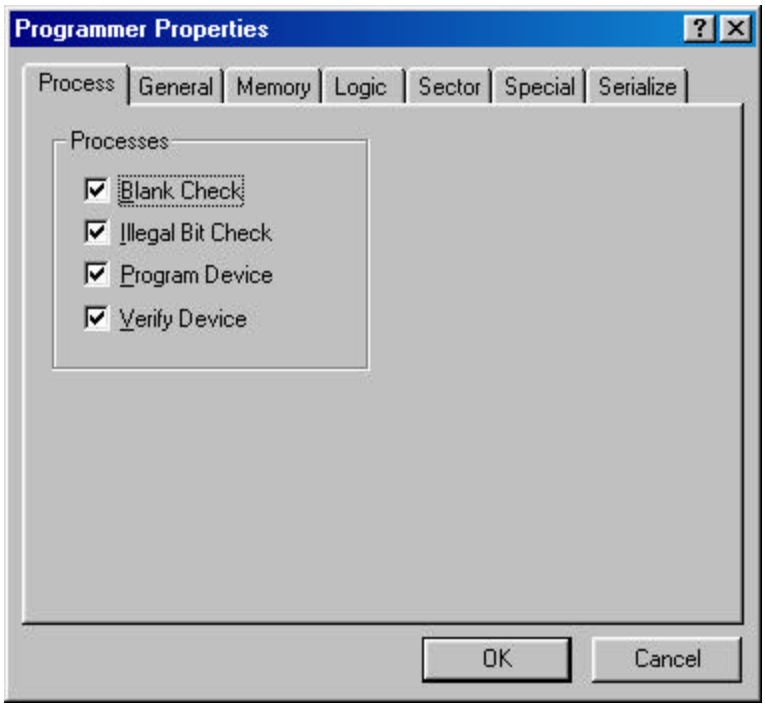
If the first non-blank line in the file does not have a colon in it, the line contents are used as the "manufacturer" name for the custom devices. If no such line is included in the file, the devices will appear under the manufacturer CUSTOM.
A custom device file for AutoSite (AUTOSITE.CUS) with a manufacturer name of ACME might look like this:

< beginning of file >
ACME
DEVICEA :
AMD 16L8-7
TI 16L8-7
NATIONAL 16L8D/7
DEVICEB : INTEL 27128
< end of file >

The spaces on either side of the colon are not required.
After running Device List Update, the first manufacturer in the Select Device list will be ACME, and the two devices, DEVICEA and DEVICEB, will be available. When ACME DEVICEA is selected, a list containing the three possible devices is presented for the operator to choose from. When ACME DEVICEB is selected, the INTEL 27128 is automatically selected.

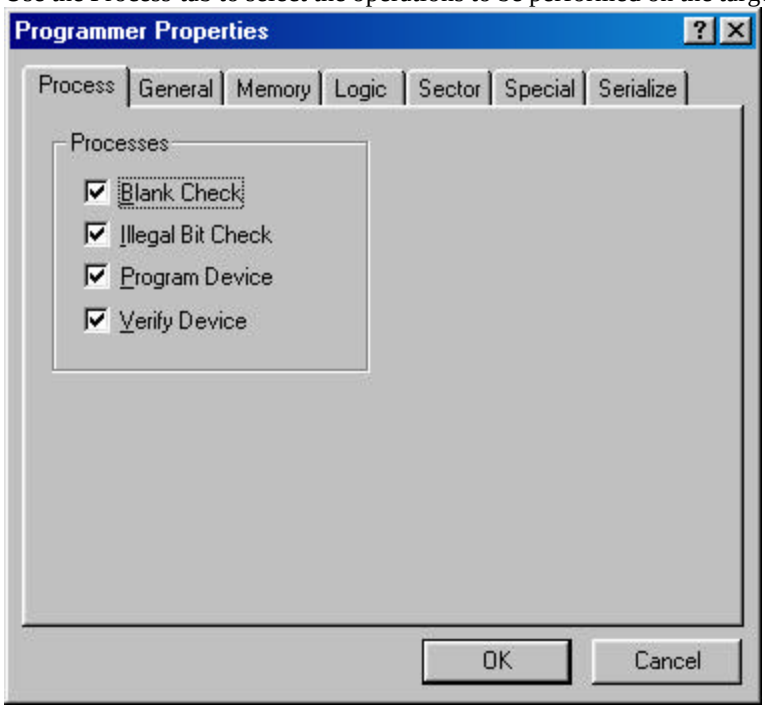
Programmer Properties

The TaskLink for UniSite Programmer Properties screen is shown below. This screen is divided into several tabs, the first one of which is for process selection.
The tabs available from Programmer Properties are Process, General, Memory, Logic, Special, Sector and Serialize.



PROCESS

The Process tab is shown below. This tab is the first tab available from the Programmer Properties screen. Use the Process tab to select the operations to be performed on the target device.



The process options include:

- Blank Check
- Illegal Bit Check
- Program Device
- Verify Device

The Select Process dialog box contains a check box for each of these operations. An X in the check box indicates that the operation is selected.

***Note**: TaskLink automatically performs a Verify operation whenever a program operation is performed. To disable the Verify operation set the verify passes to zero, in the programmer properties dialog.*

Blank Check

Use the Blank Check option to search the device(s) installed in the programmer socket(s) for programmed locations. A failure is indicated if any are found.
Selecting the Blank Check Only option from the Process Menu performs a blank check operation on the selected device. Enabling Blank Check on the Process tab of the Programmer Properties screen adds the operation to those performed when Process Devices is selected from the Process Menu.

Illegal Bit Check

An Illegal Bit Check compares device data against programmer RAM to determine if the device has programmed locations of incorrect polarity. If programmer RAM indicates that a specific bit should be in an unprogrammed state and the corresponding bit in the device is in a programmed state, a failure is indicated.
Selecting the Illegal Bit Check Only option from the Process Menu performs an illegal bit check operation on the selected device. Enabling Illegal Bit Check on the Process tab of the Programmer Properties screen adds the operation to those performed when Process Devices is selected from the Process Menu.

Program Device

Use the Program Device option to program the device(s) installed in the programmer socket(s) with the data in programmer RAM and then verify that the programmed device data matches programmer RAM. A failure is indicated if any discrepancies are found.
When the target device is a logic device, the entire fuse array is programmed. When the target device is a memory device, the following Memory Parameters affect the amount and location of data programmed:

- Begin RAM
- Begin Device
- Device Block

Selecting the Program/Verify Only option from the Process Menu performs a program/verify operation on the selected device. Enabling Program (Verify is automatically enabled also) on the Process tab of the Programmer Properties screen adds the operation to those performed when Process Devices is selected from the Process Menu.

Verify Device
Use the Verify Device option to ensure that the data in the device(s) installed in the programmer socket(s) matches programmer RAM. A failure is indicated if any discrepancies are found.
When the target device is a logic device, the entire fuse array is verified. When the target device is a memory device, the following Memory Parameters affect the amount and location of data verified:

- Begin RAM
- Begin Device
- Device Block

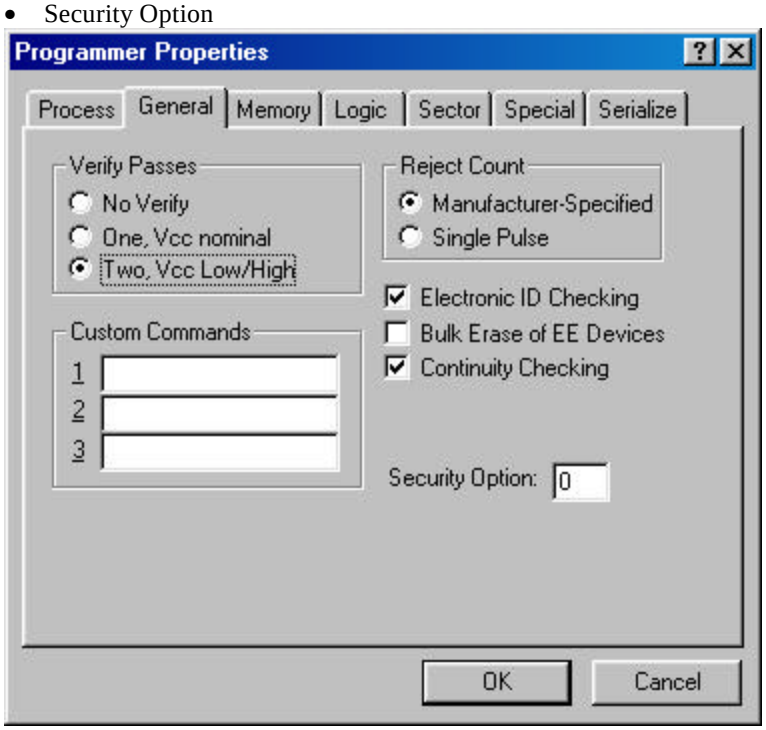
Selecting the Verify Only option from the Process Menu performs a verify operation on the selected device. Enabling Verify Device on the Process tab of the Programmer Properties screen adds the operation to those performed when Process Devices is selected from the Process Menu.

***Note**: The Verify Only function requires the "Verify Passes" parameter be set to either 1 or 2 pass. If Zero pass verify is selected, the message "Illegal Parameter Value" will appear when processing the device.*

GENERAL

The General tab is shown below. This tab is available from the Programmer Properties screen. Use the General tab to set general programming parameters. These options apply to memory, microprocessor, and programmable logic devices.
General parameters include:

- Verify Passes
- Custom Commands
- Reject Count
- Electronic ID Checking
- Bulk Erase of EE Devices
- Continuity Checking
- Security Option



***Note**: The General parameter settings are always in effect, regardless of the device type selected.*

Verify Passes
The Verify Passes options you can enter are defined below.
No Verify does not perform verification passes.
One, Vcc nominal specifies one verify pass at the nominal Vcc voltage.
Two, Vcc Low/High specifies two verify passes, one at the device's minimum Vcc voltage, and one at the device's maximum Vcc voltage.

Custom Commands
A Custom Command is a sequence of characters that TaskLink sends to the programmer before any device or data transfer operations. The Custom Command must be a valid Computer Remote Control (CRC) command for the programmer. (See your programmer's User Manual for additional information about CRC commands.) You can define up to three custom commands.
TaskLink sends a carriage return (hex 13) after the Custom Command and expects a "<" character to be returned by the programmer.
Each Custom Command may be followed by one or more flags that indicate when the command should be sent to the programmer. The flags are as follows:

P	Send command prior to device processing (program, verify, etc.)
M	Send command prior to loading master device
D	Send command prior to data download ("load RAM from file")
U	Send command prior to data upload ("create file from RAM")
E	Send command prior to RAM operations

(Fill RAM programmer command "^)
If no flags are supplied, the command is sent before any operation. The flag(s) should be placed on the Custom Command line following the command and at least one space. For example:
1 ^ D
2
3
Custom Command #1 would clear all of programmer RAM before performing a data download. RAM would not be cleared again before a programming operation.

Reject Count
The options available for the Reject Count field are:
Manufacturer-Specified - This option specifies that the programming algorithm supplies the specified manufacturer's specified number of programming pulses.
Single Pulse - This option specifies that the programming algorithm supplies a single programming pulse or a military reject count.

Electronic ID Checking
Enable the Electronic ID Checking option to verify that the Electronic ID in the device is appropriate for the selected device type. If the appropriate ID is not found, the operation is unsuccessful and a message is issued.

Bulk Erase of EE Devices
Enable the Bulk Erase of EE Devices option to cause electrically-erasable devices to be erased before being programmed with new data.

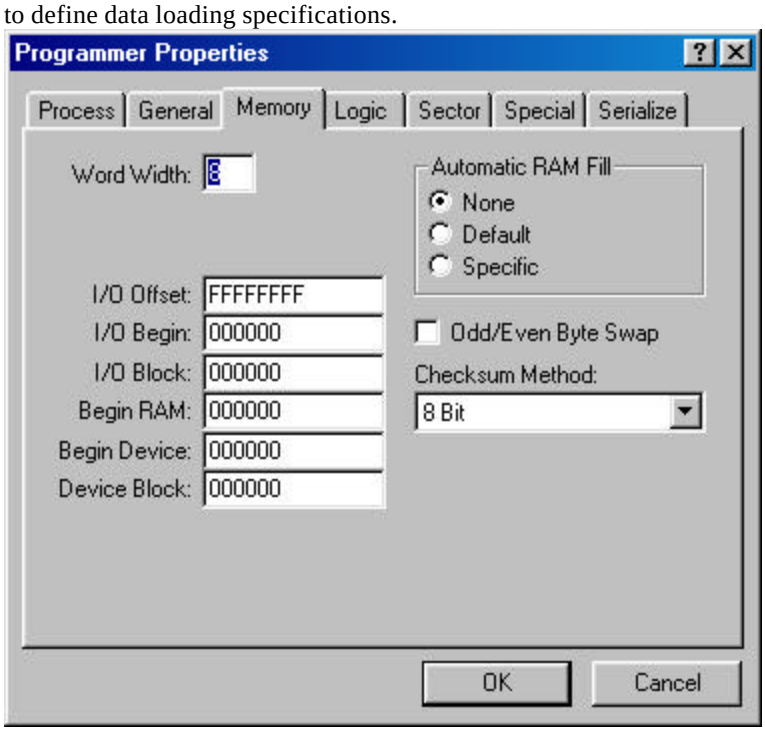
Continuity Checking
Enable the Continuity Checking option to verify that all testable device pins are making contact in the socket. If contact is not verified on all testable pins, the operation is unsuccessful and a message is issued.

Security Option
The Security Option value determines the programming of the security bit(s) in the device. The allowable range for and definition of the Security Option values differs depending on the device selected.

Option Value	Security Fuse Programming	Fuse(s)	Data
0	Disabled	All	0
1	Disabled	All	1
2	Enabled	All	0
3	Enabled	All	1
4	Enabled	First	1
5	Enabled	Second	1
6	Enabled	Third	1

MEMORY

The Memory tab is shown below. This tab is available from the Programmer Properties screen. Use this tab to define data loading specifications.



The input fields on the Memory tab include:

- Word Width
- I/O Offset
- I/O Begin
- I/O Block
- Begin RAM
- Begin Device
- Device Block
- Automatic RAM Fill
- Odd/Even Byte Swap
- Checksum Method

Word Width

The Word Width parameter sets the number of bits in the "virtual device" being programmed. The term "virtual device" is used here because setting the word width larger than the word size of the selected device actually implies the use of more than one device.

The Word Width parameter must be at least as large as the word width of the selected device and, if it is larger, it must be an integer multiple of the word width of the selected device.

I/O Offset

For Input Operations:

The address where data is placed in programmer RAM is calculated by taking the incoming address, subtracting the I/O Offset and adding the Begin RAM address. An I/O Offset of FFFFFFFF causes the I/O offset to default to the first incoming address.

For Output Operations:

The outgoing address is calculated by taking the address where the data is located, subtracting the Begin RAM address, and adding the I/O Offset. An I/O Offset of FFFFFFFF causes the I/O Offset to default to zero.

I/O Begin
The I/O Begin RAM parameter sets the first programmer RAM address from which or to which data is to be transferred in a data transfer operation.
The default value of I/O Begin RAM is 000000.

I/O Block
The I/O Block parameter sets the number of bytes to be used in data transfer operations.
The default value of 000000 causes the I/O block size to be set to the size of the selected device.

Begin RAM
The Begin RAM parameter sets the first programmer RAM address from which or to which data is to be transferred in a programmer operation.
The default value of Begin RAM is 000000.

Begin Device
The Begin Device parameter sets the first device address from which or to which data is to be transferred.
The default value of Begin Device is 000000.

Device Block
The Device Block parameter sets the number of bytes to be used in device operations.
The default value of 000000 causes the device block size to be set to the size of the selected device.

Automatic RAM Fill
The Automatic RAM Fill parameter selects the data (if any) that programmer RAM will be filled with prior to initiation of an I/O transfer to the programmer.

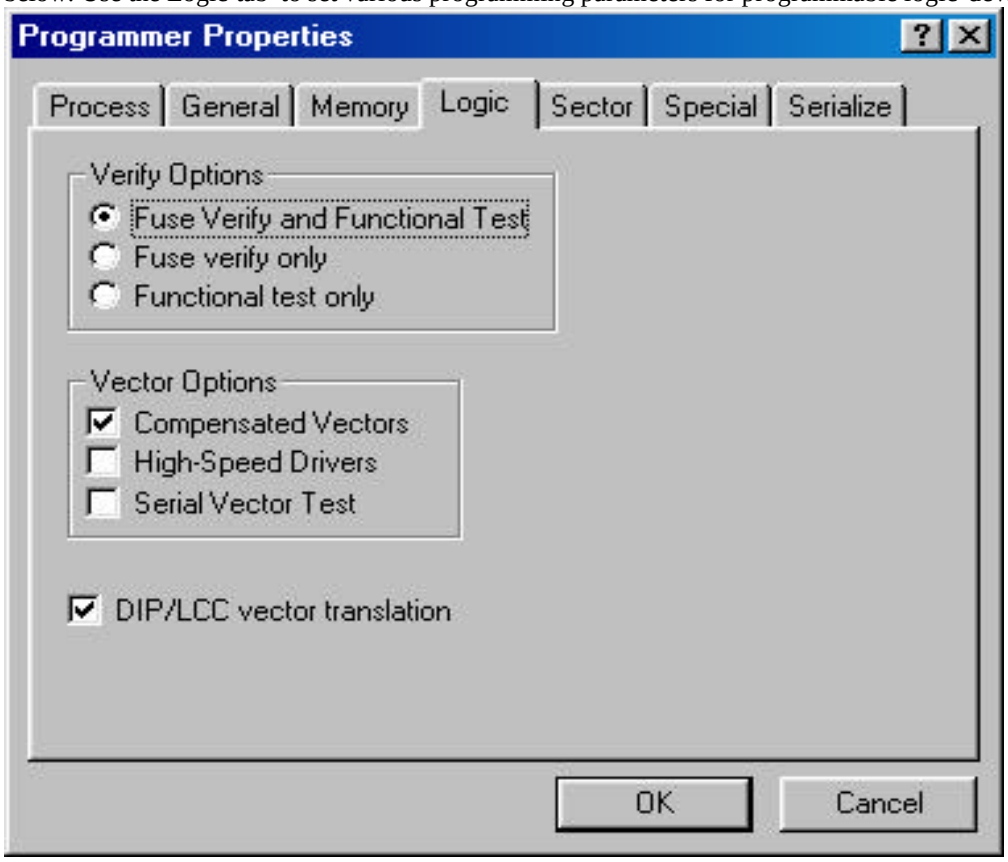
None:	Memory is not changed.
Default:	Unused locations are initialized to the unprogrammed state for the device.
Specific	Unused locations are filled with the specified two-digit hex number.

Odd/Even Byte Swap
Defines the order in which data is transferred to or from devices with a word width greater than eight bits.
When Odd/Even Byte Swap is disabled (not checked), the low byte of RAM data corresponds to the low byte (DO-D7) of the device, and the high byte of RAM data corresponds to the high byte (D8-D15) of the device.
When Odd/Even Byte Swap is enabled (checked), the low byte of RAM data corresponds to the high byte (D8-D15) of the device, and the high byte of RAM data corresponds to the low byte (D0-D7) of the device.

Checksum Method
Select a checksum method from the drop-down list. The checksum methods are 4 bit, 8 bit, and Device Word Width.

LOGIC

The Logic tab is available from the Programmer Properties screen. A picture of the Logic tab is shown below. Use the Logic tab to set various programming parameters for programmable logic devices.



Logic Parameters include:

- Verify Options
- Vector Options
- DIP/LCC Vector Translation

***Note**: The Logic Parameters settings have no effect when the target device is a memory or microprocessor device.*

Verify Options

The Verify Options parameter determines what verification procedures are performed during a verify operation.

The options are:

- Fuse verify and functional test
- Fuse verify only
- Functional test only

Functional Test usually means application of structured test vectors.

Vector Options

The Vector Options group contains the following check boxes:

Compensated Vectors provides special load compensation to the device output pins during structured vector testing. Enabling Compensated Vectors improves some vector testing problems, especially with high-speed devices.

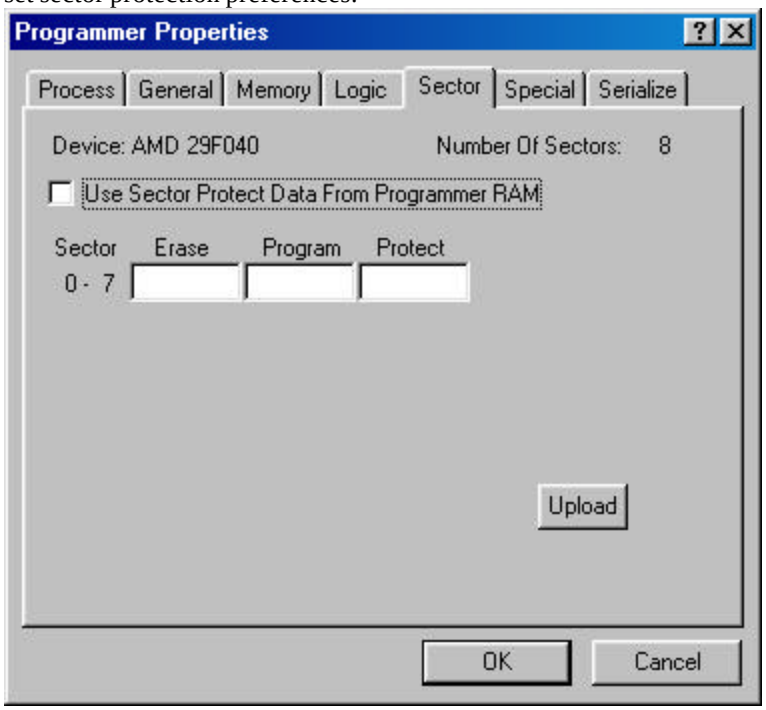
High-speed Drivers specifies a lower-impedance, faster pin driver to apply structured vector levels to device inputs. Enabling High-speed Drivers improves some vector testing problems, especially with high-speed devices.

Serial Vector Test applies the structured test vector levels serially (pin-by-pin) instead of the normal parallel method. This option provides compatibility with programmers that apply vector levels serially.

DIP/LCC Vector Translation
The DIP/LCC Vector Translation option automatically translates structured test vectors loaded from a DIP JEDEC data file to vectors appropriate for testing the PLCC version of the device.

SECTOR

The Sector tab is shown below. This tab is available from the Programmer Properties screen. Use this tab to set sector protection preferences.



The input fields on the Sector tab include:

- Device
- Number of Sectors
- Use Sector Protect Data From Programmer RAM
- Sector
- Erase
- Program
- Protect
- Upload

Device
This field displays the device selected.

Number of Sectors
The number of device sectors.

Use Sector Protect Data From Programmer RAM
When this feature is enabled, TaskLink will not send the sector parameters defined in this dialog. It is assumed that the sector data comes from a source such as a data file. Sector operations are not disabled. Any sector data or sector configuration settings enabled/disabled on the programmer are in effect.

When enabled, this feature causes TaskLink to pre-fill the dialog fields with the sector configuration from the programmer. It is assumed the data was previously sent via a source such as a data file. You can view the sector data within the data file or easily convert to the new method.
This feature is provided to give backward compatibility with older data files that may contain sector information.

When disabled, TaskLink sends all sector information defined in this dialog to the programmer, overwriting data that may have been previously sent via a data file or other means.

Sector
In this array, 8 sectors (1 row by 8 columns) are displayed. Each sector is represented by one digit. The address range for each sector depends on the size and type of the selected device.

TaskLink contacts the programmer, if it is connected, to determine the correct number of sectors for the selected device. TaskLink then displays that number (rounded to the next highest multiple of 8).

Erase
Use the Erase Sector Data fields to enable or disable sector erase operations for devices that support sector operations. Enter a 0 to disable sector erase and a 1 to enable sector erase operations.

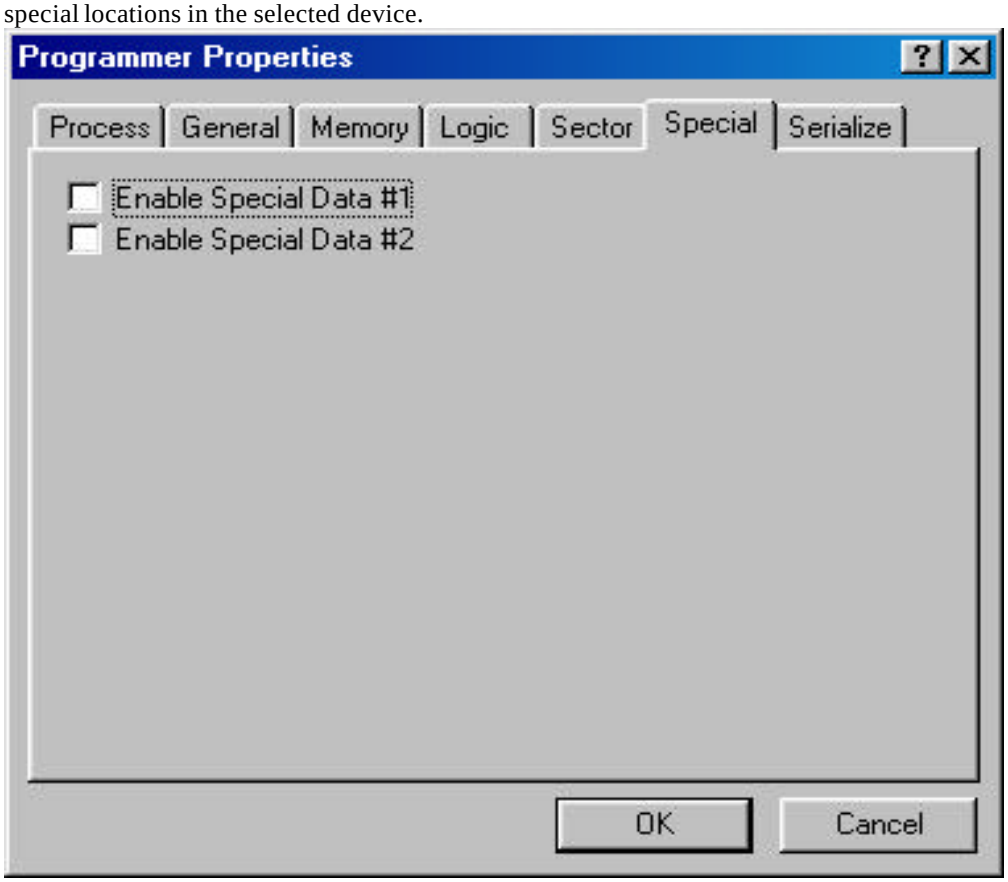
Program
Use the Program Sector Data fields to enable or disable sector programming operations for devices that support sector operations. Enter a 0 to disable sector programming and a 1 to enable sector programming operations.

Protect
Use the Protect Sector Data fields to enable or disable sector protect operations for devices which support sector operations. Enter a 0 to disable sector protect and a 1 to enable sector protect operations.

Upload
Click this button to upload data from programmer RAM.

SPECIAL

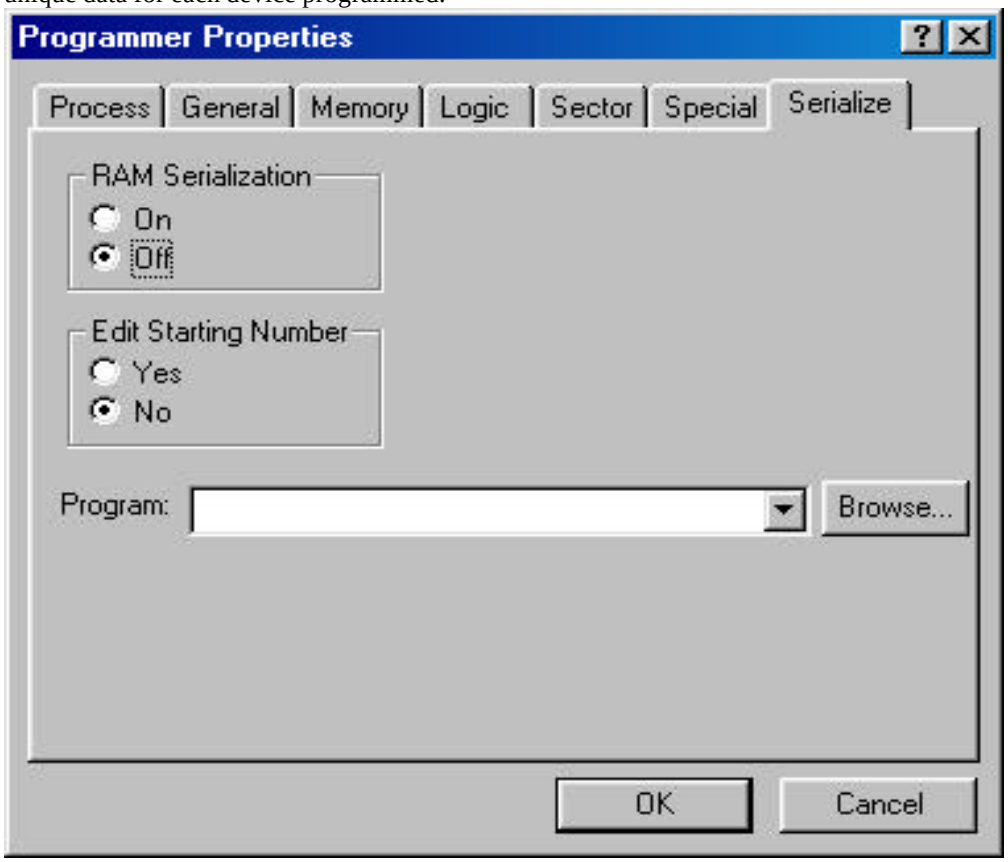
The Special tab is available from the Programmer Properties screen. A picture of the Special tab is shown below. On the Special tab, you can select **Enable Special Data #1** or **Enable Special Data #2** to program special locations in the selected device.



***Note**: The Enable Special Data option is device dependent. Refer to the Device List to determine if Enable Special Data applies to a specific device.*

SERIALIZE

The Serialize tab is available from the Programmer Properties screen. A picture of the Serialize tab is shown below. You can use this tab to set parameters associated with serialization—the generation of unique data for each device programmed.



The options available on the Serialize tab are:

- RAM Serialization ON or OFF
- Edit Starting Number YES or NO
- (Serialization) Program

RAM Serialization ON or OFF

The RAM Serialization radio buttons in the Serialize tab determine whether serial number data is placed in programmer RAM (and thus programmed into the device). Select "On" to enable RAM serialization.

Edit Starting Number YES or NO

The Edit Starting Number radio buttons in the Serialize tab determine whether or not the operator will be prompted to enter a starting serial number at the beginning of a processing session in which serialization is used. Select "Yes" to allow the operator to enter or edit the starting serial number.

Situations in which the operator might not be required to enter a starting serial number would be:

- TaskLink is storing the next number to be used in the Task record. This feature is selectable with the Keep Next Number parameter found in the Serialization dialog box when editing a Task.
- The external serialization program is responsible for keeping track of the next serial number to be used (using its own data storage).

Serialization Program

TaskLink accomplishes serialization by running an external serialization program to generate the serial number for each device. The Program entry field in the Serialization dialog box is used to specify the name of the program to be used and any command-line arguments the program requires.

Enter the name of the program along with its full path, or press the browse button for a file selection box. For example, to use the program SERIALIZ.EXE, you might enter the following:

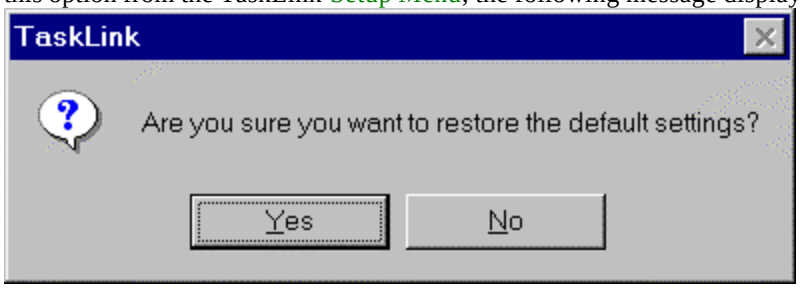
c:\dataio\tlwin\\serializ.exe -l10 -fh -a10

The flags following the program name tell SERIALIZ.EXE the address and format of the serial number to be generated. The flags must be in lower case. Upper case flags are reserved for fixed TaskLink information that is sent to the serialization program. As of version 1.70 TaskLink forces the serialization program command line to be lower case.

For more information on serializing, see SERIAL2.EXE in the System directory.

Restore Defaults

Use the Restore Defaults option to set all programming parameters to their default values. When you click this option from the TaskLink Setup Menu, the following message displays:

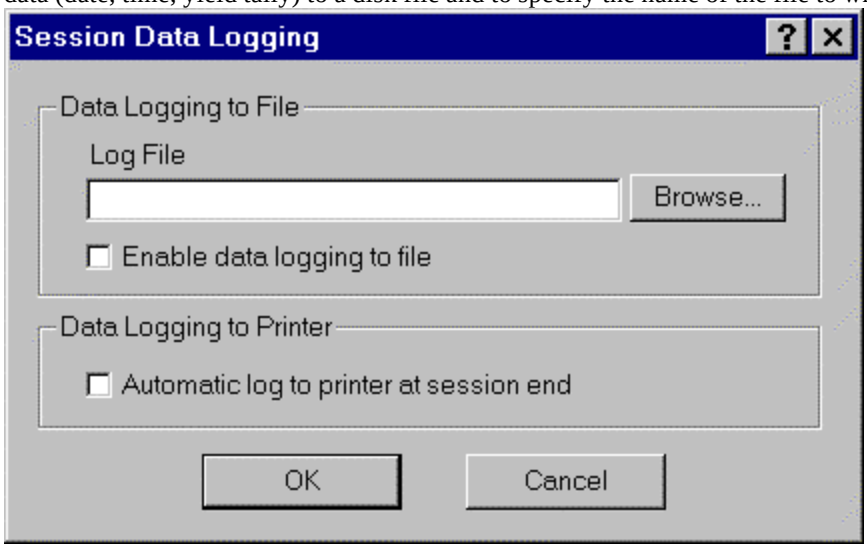


Click Yes to confirm your choice, or click No to keep your current programmer parameters.

Session Data Logging

The TaskLink Session Data Logging screen is shown below. This screen is available from the TaskLink Setup Menu.

Use the Session Data Logging option to enable or disable the logging of the programming/testing session data (date, time, yield tally) to a disk file and to specify the name of the file to which the data is logged.



***Note**: For session data logging to occur, Session Data Logging must be enabled and a log file must be specified.*

The available fields are:

- Log File
- Log File Format

LOG FILE

The Log file field is specified on the Session Data Logging screen. Use the Log File entry field to specify the file to write session data (date, time, yield tally) to at the end of a programming or testing session. To access a file selection dialog box, use the Browse button next to the entry field. You can select an existing log file or enter the name of a new log file. Log files must have a .LOG extension.

***Note**: You must enable Session Data Logging and specify a log file to use this feature.*

LOG FILE FORMAT

The log file created by TaskLink is a line-oriented ASCII file. After each programming or testing session, a ten-line record is appended to the file. An example of a TaskLink log file record is shown below followed by an explanation of each line.

Line Format

1 -
2 03-31-1992 09:47
3 Session I.D.
4 324-5678-001
5 c:\\data\\p5678001.jed
6 00007D49
7 AMD 16L8-7
8 CONT/IBC/BC/PROG/TEST/LABEL
9 357,2,0,1,0,3,5,7,0,0,1,3,0,1,3
10 XXX,XXX

Line 1: A Dash

Example: -

The first line of each record is a dash (-). This line visually separates each log file record.

Line 2: Date and Time

Example: 03-31-1992 09:47

The second line of each record contains the date and time that the record was added to the log file. The 2-digit month, 2-digit day, and 4-digit year are separated by hyphens (-).

After the date is a single space followed by the time in 24-hour format. Hours and minutes are represented by two digits, separated by a colon (:).

Line 3: Session I.D.

Example: Session I.D.

The third line of each record contains the session I.D. entered at the beginning of the programming session.

Line 4: Task Name

Example: 324-5678-001

The fourth line of each record contains the Task name that was loaded for the programming/testing session. In the example, the line contains a 10-digit part number, which is a possible Task name. If no Task is loaded during the session, this line contains a question mark (?).

Line 5: Data Source

Example: c:\\data\\p5678001.jed

The fifth line of each record contains the name of the data source used for the programming/testing session. If the data came from a PC disk file, the line contains the name of the file. Other data sources can be: Master Device, Host Download or Host: "command," where "command" is the host (remote computer) command (in quotes) used to initiate the host download. If the data source used in the session is unknown, the line will contain a question mark (?).

Line 6: Checksum of Master Data

Example: 00007D49

The sixth line of each record is the checksum of master data.

Line 7: Device Type

Example: AMD 16L8-7

The seventh line of each record contains the name of the device used for the programming/testing session. The device name appears exactly as it does in TaskLink.

Line 8: Process Selected
Example: CONT/IBC/BC/PROG/TEST/LABEL
The eighth line of each record specifies the device operations (processes) performed on the devices programmed in this session. The following abbreviations denote these operations:

- CONT Continuity test
- IBC Illegal bit check
- BC Blank check
- PROG Program
- TEST Verify

Line 9: Device Category Counts
Example: 357,2,0,1,0,3,5,7,0,0,1,3,0,1,3
The ninth line of each record contains fifteen numbers separated by commas. Each number represents the number of devices in a particular category. The position of the number (1 through 15) corresponds to the categories listed in the table below.
In the above example, there are 357 devices that passed, 2 devices with continuity fail, 0 with non-blank and so on.

Category	Example
Pass 357	Successfully programmed devices
Continuity Fail 2	Devices that have a continuity test fail.
Non-Blank 0	Devices that are non-blank prior to programming
Illegal Bit 1	Devices that contain an illegal bit
Program Fail 0	Devices that fail during programming operation
Verify Fail 1 (1 pass) 3	Devices that fail during the first verify pass
Verify Fail 2 (2 pass) 5	Devices that fail during the second verify pass
Functional Test Fail 17	Devices that fail during the first functional test pass
Functional Test Fail 20	Devices that fail during the second functional test pass
Security Fuse Fail 0	Devices with a security bit that failed to program
Security Fuse Blown 1	Devices with a programmed security bit
Device Overcurrent 3	Devices with an overcurrent error
Erase Fail 0	Electrically erasable devices that failed to erase
Electronic ID Error 1	Devices with an electronic ID that does not match the selected device type
Other Errors 3	Devices that fail due to errors other than those listed above

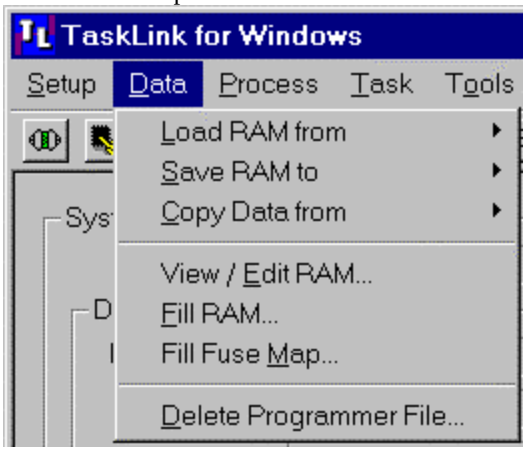
Line 10: Starting and Ending Serial Numbers
Example: XXX,XXX
The tenth line of each record contains the first and last serialization numbers of the session. If serialization is not used, only a comma (,) appears in this line.

Exit

This option is available from the TaskLink Setup Menu. Choosing Exit closes TaskLink.

Data Menu

The TaskLink Data Menu contains options for downloading data to and modifying data in programmer RAM. An example of the Data Menu is shown below.



The Data Menu includes the following options (and sub-options):

- Load RAM from PC Network File
- Load RAM from Programmer File
- Load RAM from Master
- Load RAM from Host
-
- Save RAM To PC/Network File
- Save RAM to Programmer File
-
- Copy Data from PC File to Programmer File
- Copy Data from Programmer File to PC File
- Copy Data from Master Device to Programmer File
-
- View/Edit RAM
- Fill RAM
- Fill Fuse Map
-
- Delete Programmer File

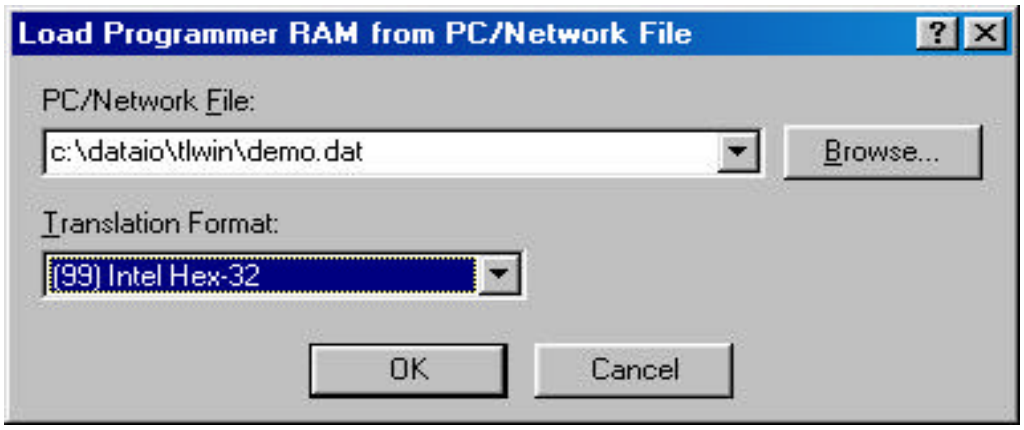
***Note**: You can disable some of the choices that involve MSMs and programmer files by going to the System Menu, selecting the Options Programmer tab and clearing the checkbox for "Enable Programmer Disk Options in Data Menu."*

Load RAM from PC/Network File

Use the Load RAM from PC/Network File option to load programmer RAM from a file either on a network or on a PC.

To load RAM from a PC or network file:

1. From the **Data Menu**, point to **Load RAM from** then click **PC/Network File**.
2. The Load RAM from PC/Network File Dialog box displays.



3. At the Load RAM from PC Disk File dialog box, enter the filename and translation format.

***Note**: When the target device is a logic device, all data in the file is loaded. When the target device is a memory device, the following Memory Parameters affect the amount and location of file data loaded:*

- I/O Offset
- I/O Block
- I/O Begin

TRANSLATION FORMAT

Use the Translation Format option to set the translation format for transferring data to and from the programmer.

With the AutoSite, UniSite, 2900, 3900, and 3980 programmers, high-speed (115.2K baud) transfer is automatically used when Translation Format is set to one of the following.

(10) Binary
(16) Absolute Binary
(95) Motorola 32-bit
(30 - 37) ASCII Octal
(50 - 58) ASCII Hex
(99) Intel Hex-32
(82) Motorola EXORciser
(87) Motorola EXORMAX
(83) Intel INTELLEC
(88) Intel MCS-86

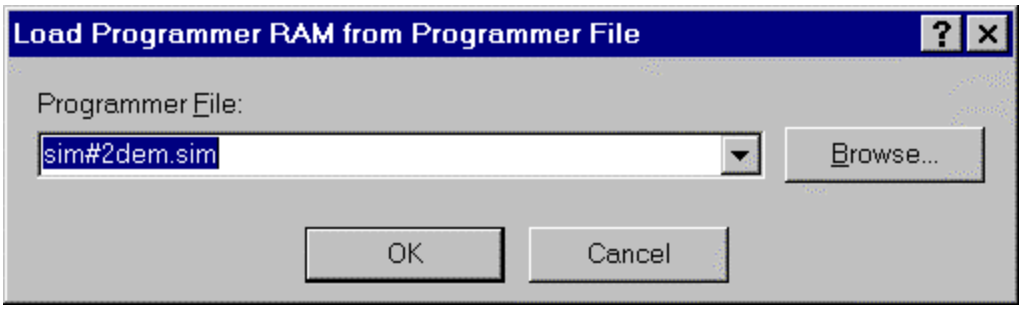
***Note**: A complete list of supported translation formats is available in the UniSite, 2900, 3900/3980, and AutoSite Translation Formats Application Note.*

Load RAM from Programmer File

Use the Load RAM from Programmer File option to load RAM from a file in the programmer.

1. From the **Data** Menu, point to **Load RAM from** then click **Programmer File.**

   The Load Programmer RAM from Programmer File dialog displays.



2. Enter the programmer filename in the field provided and click **OK**.

***Note**: When the master device is a logic device, all data in the device is loaded. When the master device is a memory device, the following Memory Parameters affect the loading of master device data:*

- Begin RAM
- Begin Device
- Device Block

Load RAM from Master

Use the Load RAM from Master option to load programmer RAM from a master device.

To load RAM from a master device:

1. From the **Data** Menu, point to **Load RAM from** then click on **Master**
2. Insert the master device to be loaded and click **OK**.

***Note**: When the master device is a logic device, all data in the device is loaded. When the master device is a memory device, the following Memory parameters affect the loading of master device data:*

- Begin RAM
- Begin Device
- Device Block

Load RAM from Host

Use the Load RAM from Host option to transfer data from a host (remote computer) and load it to programmer RAM.

To load RAM from a host:

1. Connect your host to a serial port (Host Port) on your PC. The Host Port baud rate does not have to match the baud rate of the Programmer Port.
2. From the **Data** Menu, point to **Load RAM from** then click **Host.**
3. At the Load RAM from Host dialog box, enter an optional host command and translation format.

You may need to run a terminal emulation program on the Host Port to set up the host for sending data. You can select VT100 on Host Port from the System Menu to invoke the internal VT100 emulator on the Host Port, or you can run an external program from DOS (your own terminal emulator). You can define a Task to automatically start the Host Terminal Emulator before beginning the host download.

***Note**: For the host download feature to work, the host must respond to the XON/XOFF handshaking protocol. The host must suspend transmission upon receipt of the XOFF character (hex 13) and resume transmission upon receipt of the XON character (hex 11).*

When the target device is a logic device, all host data is loaded to programmer RAM. When the target device is a memory device, the following Memory Parameters affect the amount and location of host data loaded to programmer RAM:

- I/O Offset
- I/O Block
- Begin RAM

HOST COMMAND

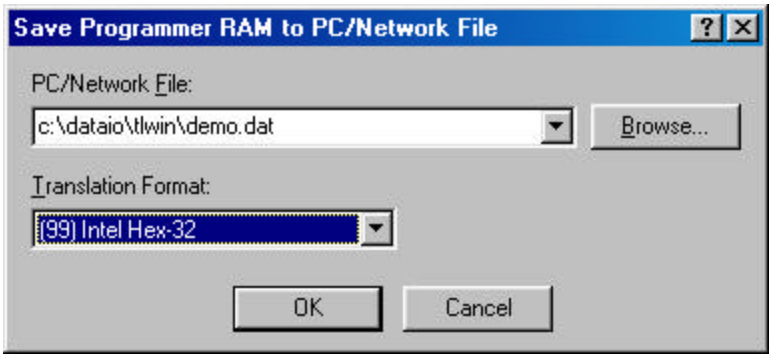
When you select the Load RAM from Host option, you are prompted for an optional Host Command: a string of characters that will be sent to the host (remote computer) to start the transmission of data. The Host Command is sent to the host when Load RAM from Host is selected. TaskLink adds a carriage return (hex 13) to the end of the Host Command.

Save RAM to PC/Network File

Use this option to save programmer RAM to a file in the selected translation format.

1. From the **Data** Menu, point to **Save RAM to…** then click **PC/Network File**.

   The Save Programmer RAM to PC/Network File Dialog displays.



2. In the dialog box, enter the filename and the Translation Format.

***Note**: When the target device is a logic device, all programmer RAM data is copied to the file. When the target device is a memory device, the following Memory Parameters affect the amount and location of programmer RAM data copied to the file:*

- I/O Offset
- I/O Block
- Begin RAM

TRANSLATION FORMAT

Use the Translation Format option to set the translation format for transferring data to and from the programmer.

With the AutoSite, UniSite, 2900, 3900, and 3980 programmers, high-speed (115.2K baud) transfer is automatically used when Translation Format is set to one of the following.

(10) Binary
(16) Absolute Binary
(95) Motorola 32-bit
(30 - 37) ASCII Octal
(50 - 58) ASCII Hex
(99) Intel Hex-32
(82) Motorola EXORciser
(87) Motorola EXORMAX
(83) Intel INTELLEC
(88) Intel MCS-86

***Note**: A complete list of supported translation formats is available in the UniSite, 2900, 3900/3980, and AutoSite Translation Formats Application Note.*

Save RAM to Programmer File

Use the Save RAM to Programmer Disk File option to save the data for the selected device from programmer RAM to a programmer's Mass Storage Module.

When you select this option, you will be prompted for the name of the programmer disk file to which the data will be written. To view a list of existing programmer disk files, press the browse button next to the filename entry field.

If you enter a programmer disk file without a drive letter, the programmer will assume drive A (a floppy disk drive). You must enter a valid drive letter to store files on the Mass Storage Module.

The data stored in programmer disk files is unformatted binary and does not have an associated translation format.

Copy Data from PC File to Programmer File

Use the Copy Data from PC Disk File to Programmer File to copy the data from a PC disk file to a programmer Mass Storage Module or a floppy disk.

When you select this option, you will be prompted for the name of the PC disk file to be read and the name of the programmer disk file to which the data will be written. To view a list of existing files, press the browse button next to the filename entry field. Since the data on the PC disk file is assumed to be formatted and the data in a programmer disk file is always unformatted binary, you must also specify a translation format.

If you enter a programmer disk file without a drive letter, the programmer will assume drive A (a floppy disk drive). You must enter a valid drive letter to store files on the Mass Storage Module.

Copy Data from Programmer File to PC File

Use the Copy Data from Programmer Disk File to PC Disk File option to copy data from the programmer's Mass Storage Module to the PC.

Upon selecting the option, you will be prompted for the name of the programmer disk file to be read and the name of the PC disk file to which the data will be written. Press the browse button next to the filename entry field to view a list of existing files. Since the data on the PC disk file is assumed to be formatted and the data in a programmer disk file is always unformatted binary, you must also specify a translation format.

If you enter a programmer disk file without a drive letter, the programmer will assume drive A (a floppy disk drive). You must enter a valid drive letter to store files on the Mass Storage Module.

Copy Data from Master Device to Programmer File

Use the Copy Data from Master Device to Programmer File option to read the data from a master device and store it in a file on the programmer's Mass Storage Module.

When you select this option, you will be prompted for the name of the programmer disk file in which the data will be stored. To view a list of existing programmer disk files, use the browse button next to the filename entry field. After specifying the filename, you will be prompted to insert the master device.

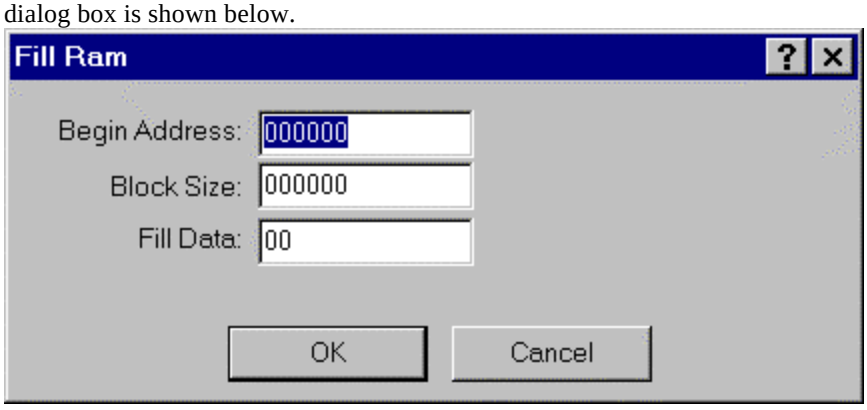
If you enter a programmer disk file without a drive letter, the programmer will assume drive A (a floppy disk drive). You must enter a valid drive letter to store files on the Mass Storage Module.
The data stored in programmer disk files is unformatted binary and does not have an associated translation format.

View/Edit RAM

Use the View/Edit RAM option to view and edit programmer RAM.
For these programmers, TaskLink uses its terminal emulator to run the programmer's internal menu system. This provides the fastest, most powerful editing capability. TaskLink for Windows uses the standard Windows application HyperTerminal as its VT100 terminal emulator.
If HyperTerminal is not installed on your PC, TaskLink will display the appropriate error message. HyperTerminal is not installed as part of the TaskLink installation. Go to Control Panel, then Add/Remove Programs, then Windows Setup tab. Highlight Communications and click on Details. Verify that HyperTerminal is checked, and then click on OK. Be ready to have your Windows CD handy depending on your installation.

Fill RAM

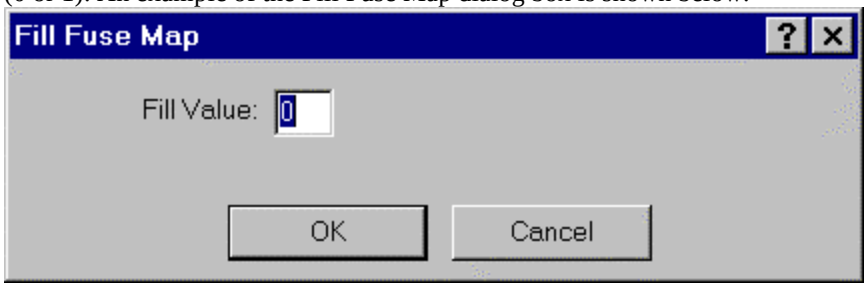
Use the Fill RAM option to fill an area of programmer RAM with a data byte. An example of the Fill RAM dialog box is shown below.



The Fill RAM dialog box has three hexadecimal entry fields:
Begin Address specifies the first location in RAM to be filled.
Block Size specifies how many bytes will be filled.
Fill Data specifies the value of the data byte to be written to RAM.
Use the Tab key to move between the fields, enter the desired values, then press Enter (or click on OK with the mouse) to begin the fill operation.

Fill Fuse Map

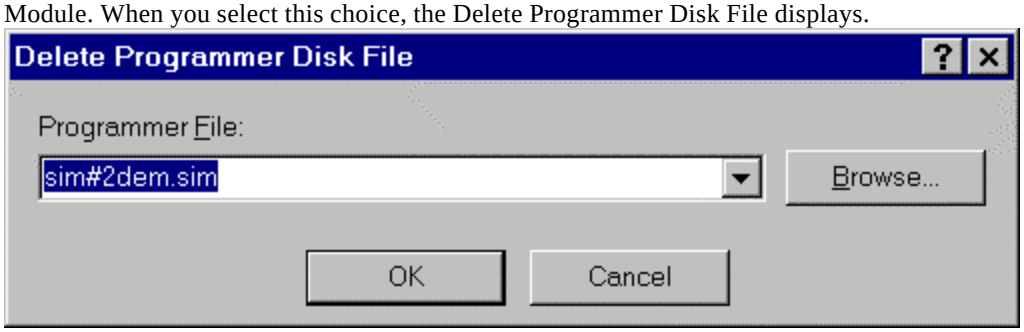
Use the Fill Fuse Map option to fill the entire array of the selected logic device with a specified fuse state (0 or 1). An example of the Fill Fuse Map dialog box is shown below.



When filling in the fuse map, "0" represents an unprogrammed state and "1" represents a programmed state.

Delete Programmer File

Use the Delete Programmer Disk File option to delete (remove) a file from the programmer's Mass Storage Module. When you select this choice, the Delete Programmer Disk File displays.

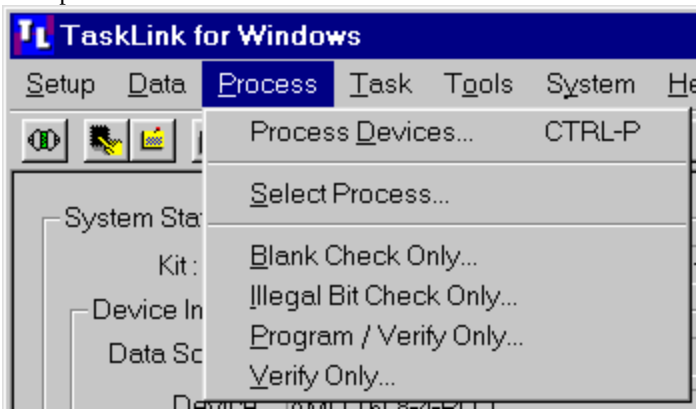


Pressing the browse button next to the filename entry field will present a list of existing programmer disk files.

If you enter a programmer disk file without a drive letter, the programmer will assume drive A: (a floppy disk drive). You must enter a valid drive letter to delete files on the Mass Storage Module.

Process Menu

The TaskLink Process Menu contains options for performing device operations together or individually. An example of this menu is shown below.



Process Menu options include:

- **Process Devices** - programs devices with the previously selected programming options.
- **Select Process**- defines programming functions to be performed on each device.
- **Blank Check Only** - searches devices for previously programmed data.
- **Illegal Bit Check Only** - -compares device data with programmer RAM to determine if device has programmed locations of incorrect polarity.
- **Program/Verify Only** - programs each device then matches the programmed data against programmer RAM.
- **Verify Only** - matches programmed data against programmer RAM to ensure they are the same.

***Note**: Some options do not apply to all programming systems.*

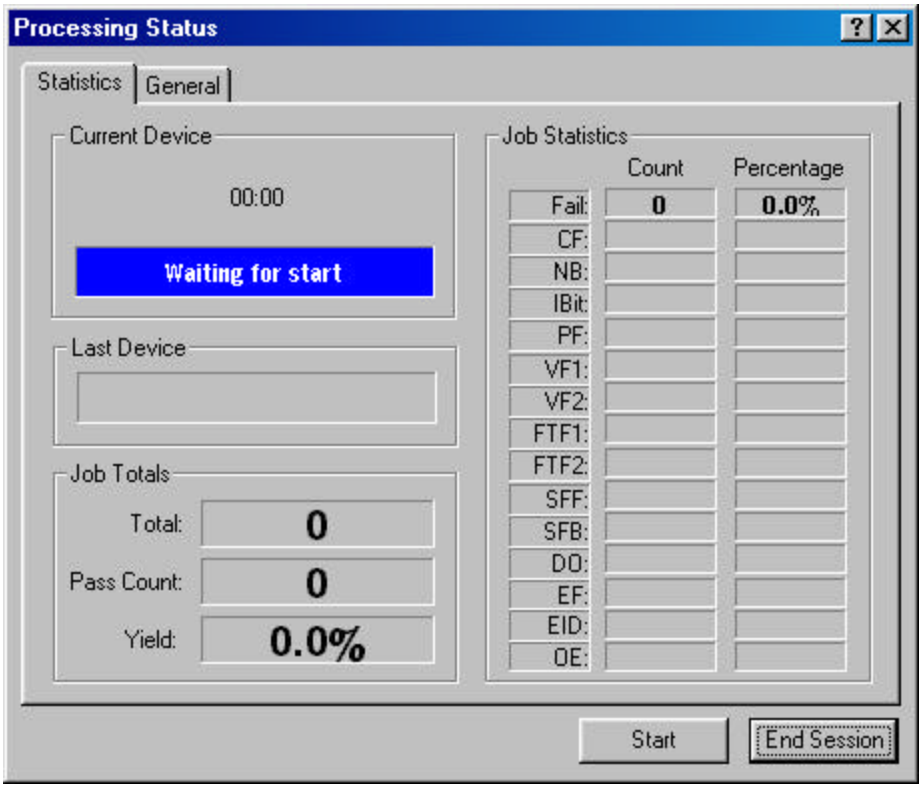
Process Devices

Use the Process Devices option to program devices with the programming operations defined in the Select Process option of the Process Menu.

To use this choice, follow the steps below:

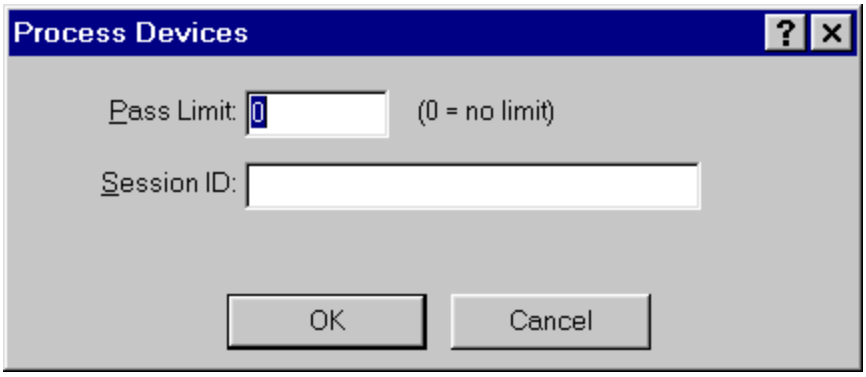
1. Choose Process Devices from the Process Menu.

The Processing Status screen displays. The Statistics tab gives you pass/fail statistics on your last job. The General tab shows you other processing parameters.



2. Click OK.

The Process Devices dialog displays.



3. Enter a Pass Limit and an optional Session ID.

If any device footnotes apply to the devices you are processing, they will display.

4. Click Start on the Processing Status screen.

TaskLink will process devices until the pass limit you specified has been reached.

Select Process

Use the Select Process option to select the operations to be performed on the target device.
The process options include:

- Blank Check
- Illegal Bit Check
- Program Device
- Verify Device

The Select Process dialog box contains a check box for each of these operations. An X in the check box indicates that the operation is selected.

***Note**: TaskLink automatically performs a Verify operation whenever a program operation is performed. To disable the Verify operation set the verify passes to zero, in the programmer properties dialog.*

Blank Check Only

Use the Blank Check option to search the device(s) installed in the programmer socket(s) for programmed locations. A failure is indicated if any are found.
Selecting the Blank Check Only option from the Process Menu performs a blank check operation on the selected device. Enabling Blank Check on the Process tab of the Programmer Properties screen adds the operation to those performed when Process Devices is selected from the Process Menu.

Illegal Bit Check Only

An Illegal Bit Check compares device data against programmer RAM to determine if the device has programmed locations of incorrect polarity. If programmer RAM indicates that a specific bit should be in an unprogrammed state and the corresponding bit in the device is in a programmed state, a failure is indicated.
Selecting the Illegal Bit Check option from the Process Menu performs an illegal bit check operation on the selected device. Enabling Illegal Bit Check on the Process tab of the Programmer Properties screen adds the operation to those performed when Process Devices is selected from the Process Menu.

Program/Verify Only

Use the Program/Verify option to program the device(s) installed in the programmer socket(s) with the data in programmer RAM and then verify that the programmed device data matches programmer RAM. A failure is indicated if any discrepancies are found.
When the target device is a logic device, the entire fuse array is programmed. When the target device is a memory device, the following Memory Parameters affect the amount and location of data programmed:

- Begin RAM
- Begin Device
- Device Block

Selecting the Program/Verify Only option from the Process Menu performs a program/verify operation on the selected device. Enabling Program (Verify is automatically enabled also) on the Process tab of the Programmer Properties screen adds the operation to those performed when Process Devices is selected from the Process Menu.

Verify Only

Use the verification to ensure that the data in the device(s) installed in the programmer socket(s) matches programmer RAM. A failure is indicated if any discrepancies are found.
When the target device is a logic device, the entire fuse array is verified. When the target device is a memory device, the following Memory Parameters affect the amount and location of data verified:

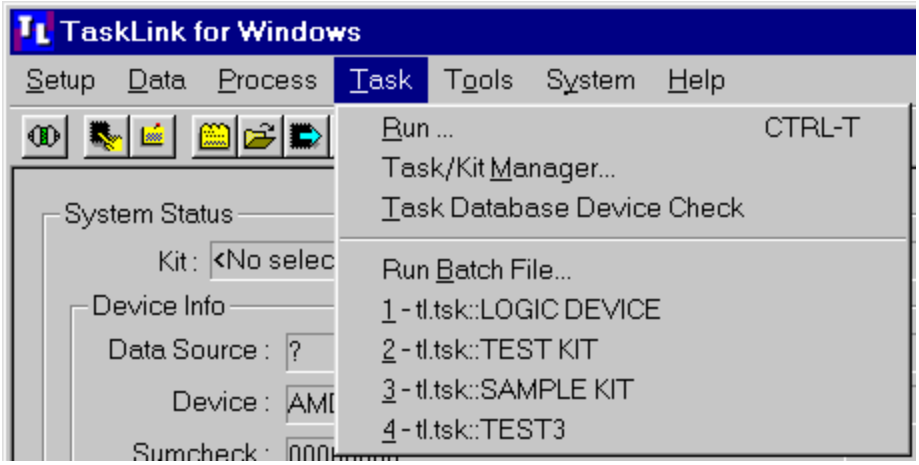
- Begin RAM
- Begin Device
- Device Block

Selecting the Verify Only option from the Process Menu performs a verify operation on the selected device. Enabling Verify on the Process tab of the Programmer Properties screen adds the operation to those performed when Process Devices is selected from the Process Menu.

***Note**: The Verify Only function requires the "Verify Passes" parameter be set to either 1 or 2 pass. If Zero pass verify is selected, the message "Illegal Parameter Value" will appear when processing the device.*

Task Menu

The TaskLink Task Menu contains options to add, remove, and edit Tasks and Kits. An example of the Task Menu is shown below.



Tasks and Kits allow you to define all of the information needed to perform a single programming job (a Task) or a sequence of programming jobs (a Kit). Tasks and Kits are the two types of data records found in a Task/Kit Database file. A Task/Kit database can contain up to 5000 records, which can be only Tasks, only Kits, or a combination of both.

Task Menu options include:

- Run
- Task/Kit Manager
- Task Database Device Check
- Run Batch File
- Run Last Task/Kit (listed by filename)

See also:

What Are Tasks and Kits

Task Database Device Check

Use the Task/Kit Database Device Check option to verify that the device names stored in the selected Task Database file are supported in the current device support files. When an unrecognized device name is found in a Task, the Select Device dialog box appears prompting you to select a replacement device. TaskLink then searches the rest of the Tasks in the Task Database file, replacing each occurrence of the device. Occasionally, a programmer update includes a device name change (for example, an ACME 16L8A might become an ACME 16L8A/B). If a Task contains the name of a specific device whose name was changed in the programmer's device support list, TaskLink displays a "Device not Found" message when the Task is loaded.

***Note**: Using wildcards (such as * *16L8*) when creating Tasks can reduce device support update conflicts.*

Run Batch File

Use the Run Task/Kit Batch File option to execute a group of Tasks and/or Kits as defined in an ASCII batch file.

The batch file can be created with any DOS text editor. Each line of the batch file should contain the name of a Task or Kit to be executed. The Task or Kit name can optionally be preceded by the Task/Kit Database filename and a double colon (::). If the Task/Kit Database filename is not included, TaskLink will attempt to find the Task in the last Task Database used.

For example, to run task_name in a specific Task database file, enter:

task_database_filename::task_name

To run task_name in the default Task database file, enter:

task_name

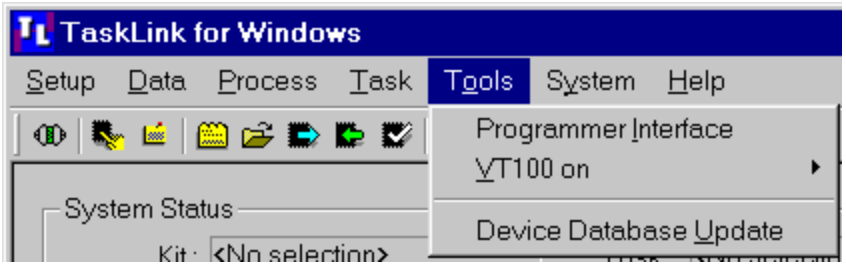
***Note**: Do not use two consecutive colons in Task or Kit names if you plan to use batch files.*

Run Last Task/Kit (listed by filename)

To shortcut the Run dialog, click the filename of the last Task or Kit run from the Task Menu and TaskLink will rerun the process as previously programmed without any additional prompting.

Tools Menu

The TaskLink Tools Menu options allow you to use programmer menu commands via terminal emulation, set up terminal emulation, and update your device database. These options rely completely on the communication link you have established with the programmer. An example of the Tools Menu is shown below.



The options on the Tools Menu are listed below:

- Programmer Interface
- VT100 on Programmer Port
- VT100 on Host Port
- Device Database Update

Programmer Interface

Use the Programmer Interface option to enter the programmer's menu system and access programmer functions not available with TaskLink. TaskLink for Windows uses the standard Windows application HyperTerminal as its VT100 terminal emulator.
If HyperTerminal is not installed on your PC, TaskLink will display the appropriate error message. HyperTerminal is not installed as part of the TaskLink installation.

To install HyperTerminal:

1. From the Windows Control Panel, click Add/Remove Programs, then choose the Windows Setup tab.
2. Highlight **Communications** and click **Details**.
3. Verify Hyper Terminal is checked, then click on **OK**, then **OK**.
4. Be ready to have your Windows CD handy depending on your installation.
5. To return to TaskLink, select **EXIT** from the File Menu.

VT100 on Programmer Port/Host Port

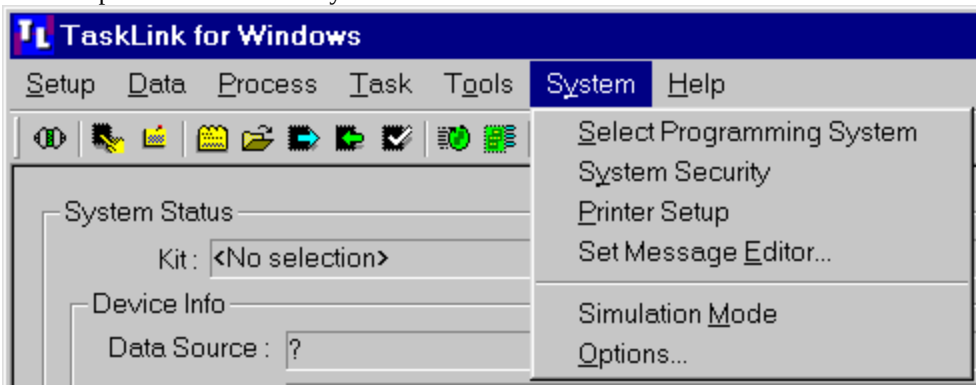
TaskLink for Windows uses the standard Windows application HyperTerminal as its VT100 terminal emulator.
The VT100 emulator is accessible via the VT100 on Programmer Port and VT100 on Host Port options in the Tools Menu.
The VT100 emulator uses the port settings set in the Options/ Programmer Port and Options/Host Port dialog boxes.

Device Database Update

Use the Device Database Update option to update the TaskLink programmer device support files after a programmer update is installed. TaskLink uploads the device list from the programmer and saves it to a PC disk file.
When Device Database Update is selected, a window appears showing the number of devices uploaded so far and the name of the manufacturer whose devices are currently coming in. This process may take several minutes, depending on the number of devices supported by the programmer.

System Menu

The TaskLink System Menu contains options for TaskLink system configuration that allow you to configure your individual programming system.
An example of the TaskLink System Menu is shown below.

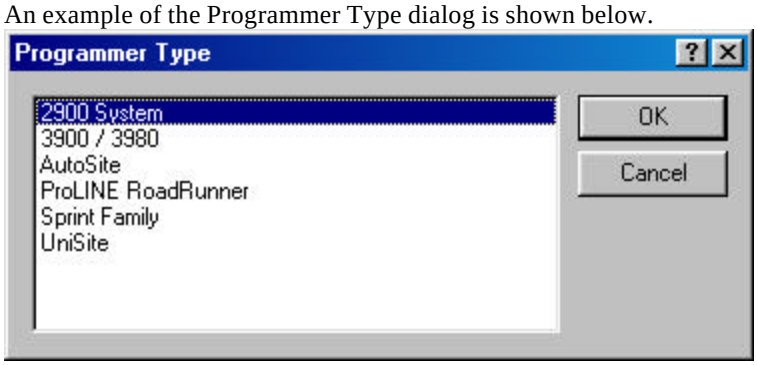


The System Menu options are:

- Select Programming System
- System Security
- Printer Setup
- Set Message Editor
- Simulation Mode
- Options

Select Programming System

Use the Programmer Type option to specify the Data I/O programmer controlled by TaskLink.
An example of the Programmer Type dialog is shown below.



The Programmer Type dialog box contains a list of supported programmers. When you select a programmer from the list, TaskLink is configured based on the features available on the selected programmer and loads the programmer messages from disk.

System Security

Use the Security Setup option to define the level of security (user restriction) that TaskLink provides and the information (usernames, passwords, and authorization levels) necessary for the implementation of security features.
TaskLink has two modes of operation--Operator mode, in which users can only run pre-defined Tasks, and Administrator mode, in which users can create, edit and delete Tasks as well as use any of the features available in the main menu of TaskLink.
TaskLink provides three selectable levels of security:

- No security.

When “No Security” is enabled, Administrator mode is entered when starting TaskLink. All operations in TaskLink are available for use in Administrator mode.

- Single administrator-mode password.

When “Single administrator mode password” is enabled, Operator mode is entered when starting TaskLink, and the Run Task/Kit screen is displayed. You can enter Administrator mode by pressing F10 while in the operator mode dialog box and entering the password when prompted.

- Defined users with individual passwords and authorization levels.

When “Defined users with individual passwords and authorization levels” is enabled, TaskLink begins and the user is prompted to enter Name and Password (which they can change at any time). Users can be defined as either an operator or an administrator. TaskLink will automatically start up in the mode defined for that user. From operator mode, administrator mode may be entered pressing F10 while in the operator mode dialog box.
When the security level is set to anything except "No Security," a password is required to enter the Security Setup dialog box. This password can be changed from within the Security Setup dialog box.

USERNAME

Use this field to enter the name of a new TaskLink user.

PASSWORD

Use this field to enter the password for a new TaskLink user.

Printer Setup

The Printer Setup dialog allows you to view the current printer driver settings, and to change printer types. All TaskLink features that print, print to the currently selected Windows printer. At this time, TaskLink does not allow the user to specify and store local printer settings. TaskLink only uses the default printer driver settings. In order to change the print setting, you will need to change the printer driver default settings.
TaskLink prints in a generic text format and does not use any special formatting. TaskLink should be able to print to any printer supported by your Windows operating system.

Set Message Editor

Use the Set Message Editor option to establish the text editor program that TaskLink will use for editing the file TL.MSG.
TL.MSG contains the (optional) informative messages that are displayed for the operator at the beginning of each Task when running a Kit. The specified message editor is run (with the filename TL.MSG as a command-line argument) when the Message Editor is activated from the Edit Kit dialog box. TaskLink does not search the path defined in the DOS PATH variable, so make sure to enter the full path description for your editor. For example, to use the text editor that comes with Windows, you might enter C:\windows\notepad.exe.
The purpose of the Message Editor is to provide quick access to a text editor to compose or view the messages associated with the Kit you are editing. Use of this feature is entirely optional-- the file TL.MSG can be edited at any time outside of the TaskLink program.

CREATE MESSAGES

When in the Edit Kit dialog, selecting **Msg Editor** allows you to create (optional) informative messages that are displayed for the operator at the beginning of each Task when running a Kit.
This button invokes the text editor specified in the Set Message Editor option in the System Menu. The editor is passed the name of the file TL.MSG (and any path specified in the TLFILES environment variable).
Once in the text editor, each message must be assigned a number that follows this format:
~ followed by four digits
Examples of text message numbers:
~0001 ~0012 ~0254 ~3578
After assigning a number, add a carriage return before typing the text of the optional message.
A sample message might look like this:
~0024
Run this Kit before any others.

Simulation Mode

Simulation mode is the TaskLink demonstration mode accessed from the System Menu. Simulation mode allows you to run most TaskLink functions without actually having a Data I/O programmer connected. Some functions cannot be performed in simulation mode and will cause the message "Operation not allowed in Simulation Mode" to appear.

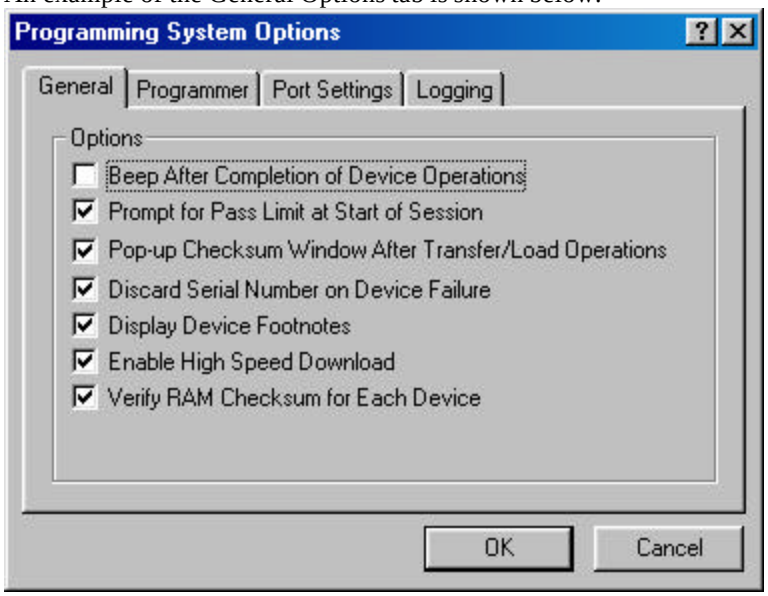
***Note**: When processing devices in simulation mode, TaskLink generates device failure conditions randomly. These failures may not be appropriate for the operation(s) being simulated. This is normal and is characteristic of simulation mode only.*

Options

When you select Options from the System Menu, the Programming System Options window displays. The Programming System Options window is divided into four tabbed dialog boxes:

- General
- Programmer
- Port Settings
- Logging

An example of the General Options tab is shown below.



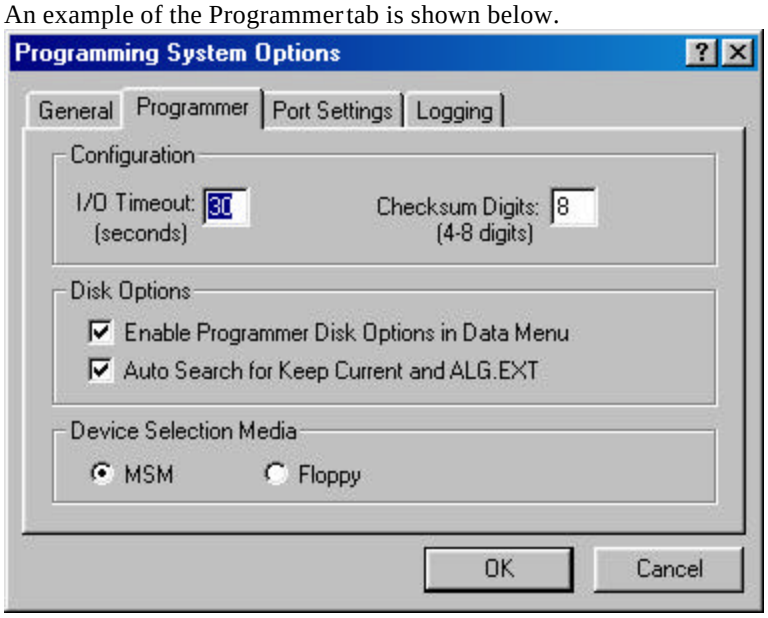
GENERAL

Use the General Options tab to select your program behavior preferences. Only preferences applicable to the configuration you selected will appear on the tab.
The available options are:

- Beep After Completion of Device Operations
- Prompt for Pass Limit at Start of Session
- Pop-up Checksum Window After Transfer/Load Operations
- Discard Serial Number on Device Failure
- Display Device Footnotes
- Enable High-Speed Download
- Verify RAM Checksum for Each Device

PROGRAMMER

The Programmer tab of the Programming System Options dialog contains options that allow you to configure the storage and set the parameters for your programmer.
An example of the Programmer tab is shown below.



The options available on the Programmer tab are listed below.
I/O Timeout - Sets the amount of time you want the programmer to wait for incoming data to arrive during a data input operation before flagging an error.
I/O Timeout may be useful when downloading data from a host (remote computer) manually or from a host that takes a long time to respond to the data request. Setting the I/O Timeout to zero causes the programmer to wait indefinitely.

Checksum Digits - Sets the maximum number of digits TaskLink will use to display programmer RAM checksums on the screen. The Checksum Digits dialog box contains one entry field. Enter the desired maximum number of checksum digits (4 to 8).

Enable Programmer Disk Options in Data Menu - causes TaskLink to present options (in the Data pull-down menu) relating to data storage in programmer disk files (floppy drive or Mass Storage Module). Since the addition of the programmer disk options to the Data Menu significantly increases the number of choices in the menu, you may wish to disable these options if you do not plan to store data in programmer disk files.

Auto Search for Keep Current and ALG.EXT - allows TaskLink to automatically search for Keep Current Algorithms and Extended Algorithms. TaskLink will ignore the Device Selection Media setting

and search the floppy drive first, then the MSM. With this feature disabled TaskLink will use the Device Selection Media setting for selecting Keep Current and Extended Algorithms.

Device Selection Media - When you enable MSM as the Device Selection Media, TaskLink always selects standard device algorithms from the specified media.

PORT SETTINGS

Use the Port Settings dialog to set up the parameters for the Programmer or Host ports.
The Port Settings dialog box has a group of combo-boxes for each parameter. The Programmer Port settings must match the settings of the programmer's serial port. For optimum performance, the ports should be set to the fastest baud rate supported by the programmer.
Recommended Port settings are:

Port	COM1 or COM2
Baud Rate	9600 (19200 if supported)
Parity	None
Data Bits	8
Stop Bits	1

Following is a brief explanation of each parameter:

Port

Selects which PC serial port is being used to connect to the programmer. COM1 and COM2 are available on most PCs.

Baud Rate

Selects the baud rate. Baud rate is the rate (bits per second) at which data bits are transmitted and received across the serial connection. For 8-bit data with one start and one stop bit and no parity bit, the rate at which characters are sent is equal to the baud rate divided by 10. For example, 9600 baud represents a character transmission rate of 960 characters per second.

Parity

Selects the kind of parity. Parity is an error detection method in which the sending port adds an extra bit to each character to force the number of 1s in the character and parity bit combination to be even (even parity) or odd (odd parity). If the receiving port detects an incorrect parity in the received character, an error is flagged. Select None to disable parity checking.

Data Bits

Selects the number of data bits. Data bits are the bits representing the actual data (character) being transmitted. All binary data translation formats and the built-in RAM data editor in TaskLink use 8 data bits.

Stop Bits

Selects the number of stop bits. Stop bits are the bits used to signal the end of a serial data character.

LOGGING

The Logging tab is used to enable and set parameters for TaskLink's COM Logging feature.
The COM log is used as a troubleshooting tool by users and by Data I/O technical support and engineers. When the COM log is enabled all commands sent back and forth between the programming system and TaskLink are logged into the specified file. The log file is typically e-mailed to Data I/O technical support engineers for review. This allows Data I/O engineers to determine if problems are related to communications between systems without having to travel to the customer site or hooking up expensive analyzers.

Help Menu

The TaskLink Help Menu provides access to topics that describe the operation of TaskLink and the Help system.
The Help Menu options are:

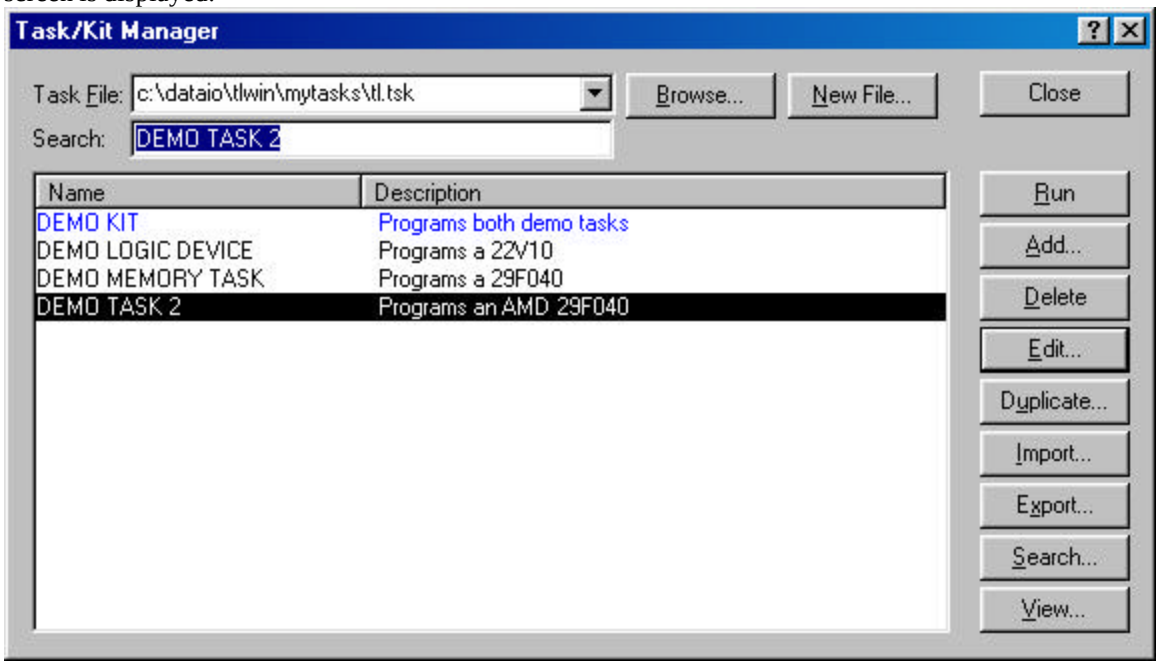
- **Help Topics** - Used for accessing TaskLink Help
- **Technical Support**- TaskLink technical support numbers
- **On-line Documentation** – Used for accessing TaskLink documentation
- **Data I/O on the Web** - Direct access to Data I/O web sites
- **About TaskLink -** TaskLink version information

TASKLINK: UNISYSTEM OPERATIONS

TaskLink offers users the option of creating Tasks and Kits to streamline device programming. Here is some important information about creating and changing Tasks and Kits using TaskLink.

First, you must be in Administrator Mode to create or change Tasks or Kits. If you entered TaskLink in Operator Mode, you are only allowed to run Tasks or Kits, not create or change them. To enter Administrator Mode from Operator Mode, click on Admin-F10 and enter your Password.

Once you're in Administrator Mode, select Task/Kit Manager from the Task Menu. The Task/Kit Manager screen is displayed:

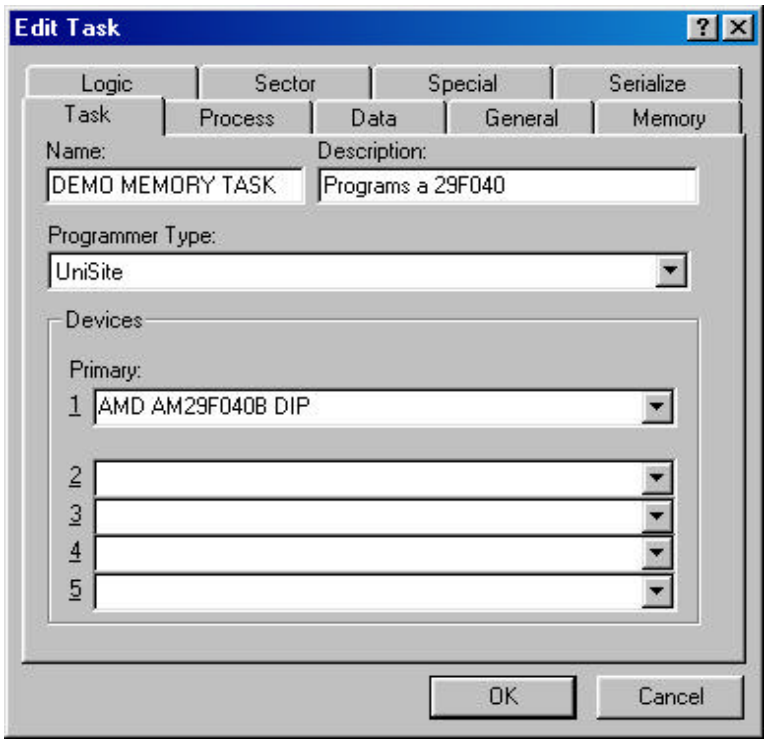


You'll notice that you have many choices of operations to perform once in the Task/Kit Manager screen. Here are the most frequently used operations:

- Run (a Task or Kit)
- Edit (a Task or Kit)
- View (a Task or Kit)

Edit a Task

The Edit Task screen is available when you highlight a task and click Edit from the Task/Kit Manager. Use the Edit Task screen to change task parameters. An example of the Edit Task screen is shown below.



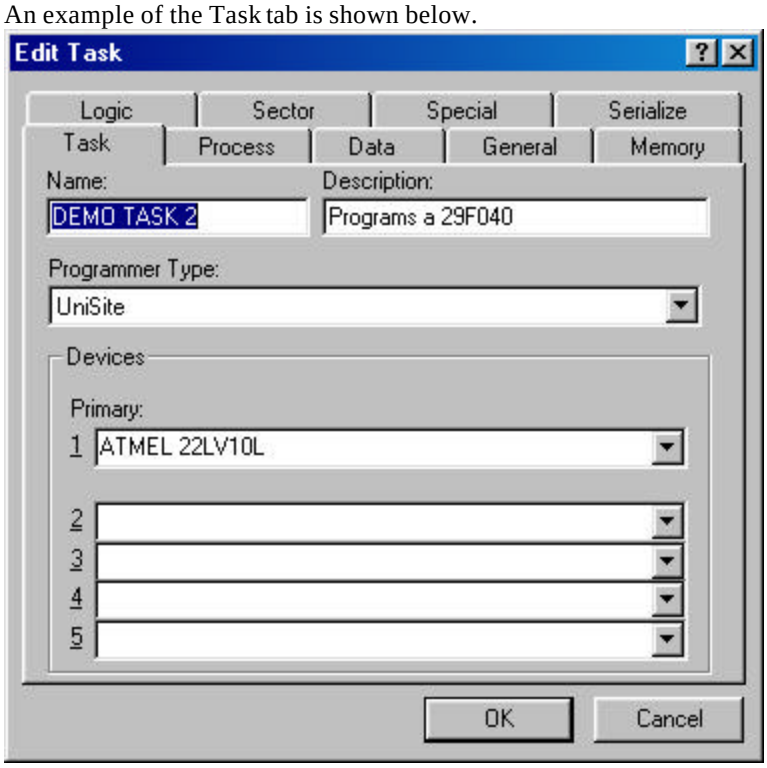
Task parameters are located in tabs on the Edit Task screen. These parameters include:
Task - details Task name and description, programmer type, processes and devices to be programmed.
Data - contains information on data sources.
General - contains selections on number of Verify Passes, Reject Count type, Custom Commands, Electronic ID Checking, Bulk Erase of EE Devices and Continuity Checking.
Memory - contains size and input options for memory.
Logic - contains verify and vector options.
Sector - for use in only some devices, these parameters allow for erasing, programming and protecting individual sectors.
Special - permits special data to be entered.
Serialize - enables the serialization process.

Task

The Task tab available from the Edit Task screen includes the following utilities:
Name and Description - the name and description for the programmed task.
Programmer Type - displayed as a dropdown list, choose from AutoSite, UniSite, Sprint Family, RoadRunner, 2900 or 3900/3980
Devices - Choose device model and manufacturer for up to five different devices.

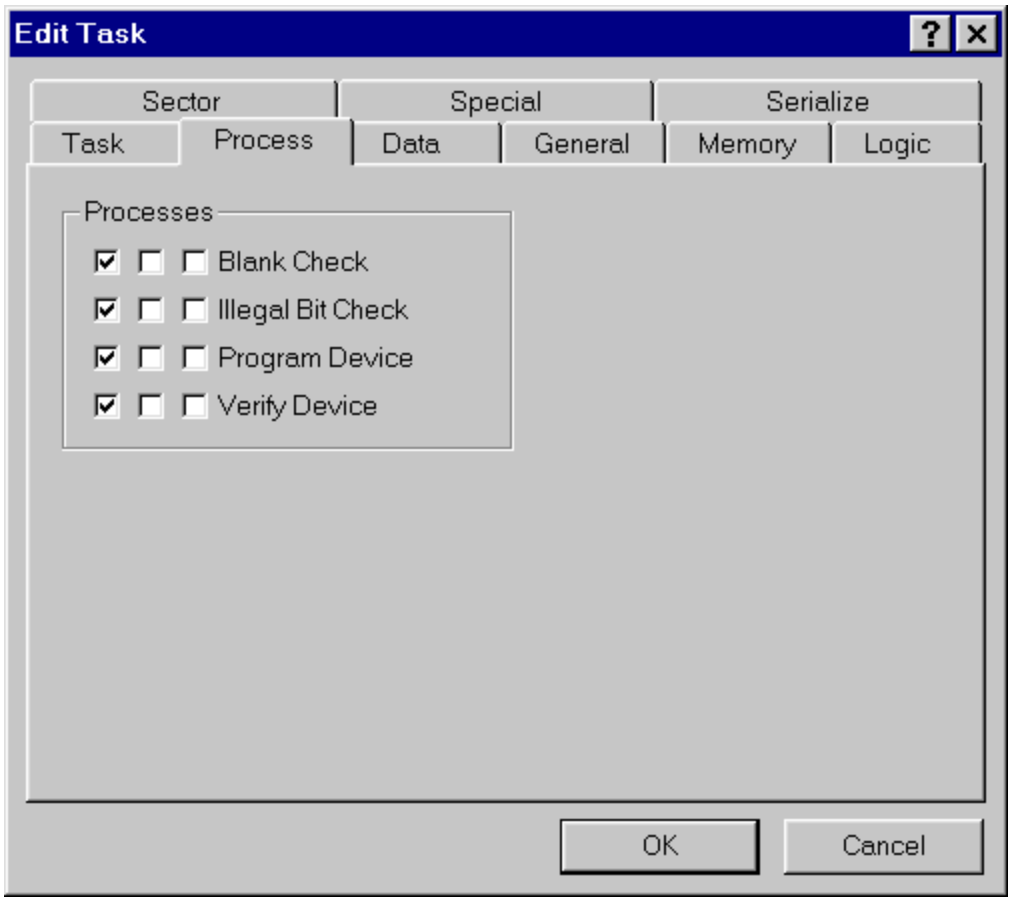
***Note**: The device you select for the #1 Primary field serves to set the available options for programming operations. You may not enter a wildcard for the #1 Primary device. For devices #2-5, you may enter wildcards. If you specify more than one device or at least one wildcard, TaskLink displays a list of devices to choose from when the Task is loaded.*

An example of the Task tab is shown below.



Process

The Process tab available from the Edit Task screen is shown below.



Options on the Process tab include:

- Blank Check
- Illegal Bit Check
- Program Device
- Verify Device

Blank Check

Use the Blank Check option to search the device(s) installed in the programmer socket(s) for programmed locations. A failure is indicated if any are found.

Illegal Bit Check

An Illegal Bit Check compares device data against programmer RAM to determine if the device has programmed locations of incorrect polarity. If programmer RAM indicates that a specific bit should be in an unprogrammed state and the corresponding bit in the device is in a programmed state, a failure is indicated.

Program Device

Use the Program Device option to program the device(s) installed in the programmer socket(s) with the data in programmer RAM and then verify that the programmed device data matches programmer RAM. A failure is indicated if any discrepancies are found.

Verify Device

Use the Verify Device option to ensure that the data in the device(s) installed in the programmer socket(s) matches programmer RAM. A failure is indicated if any discrepancies are found.

Data

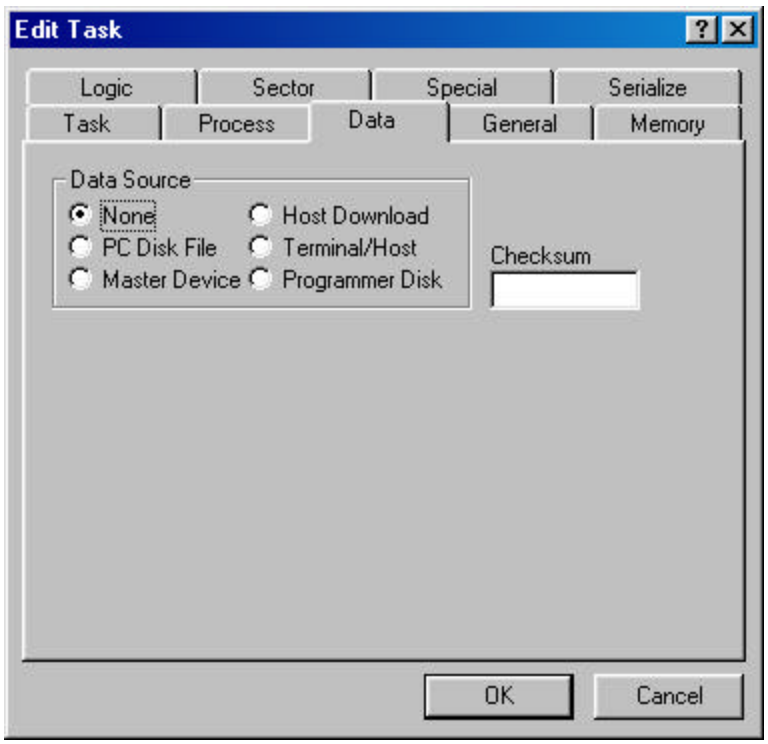
The Data tab available from the Edit Task screen includes utilities for:

Data Source - select from PC Disk File, Master Device, Host Download, Terminal/Host and Programmer Disk

- PC Disk File – Select the PC Disk File option to download data from a PC or disk file.
- Master Device – Select the Master Device option to download data from a master device.
- Host Download – Select the Host Download option to download data from a host computer.
- Terminal/Host – Select the Terminal/Host option to download data from a terminal or host computer.
- Programmer Disk – Select the Programmer Disk option to download data from a programmer disk.

Checksum – Enter a checksum value that TaskLink will compare to the checksum value in Programmer RAM. If the two values don't match, an error message will be displayed.

An example of the Data tab is shown below.

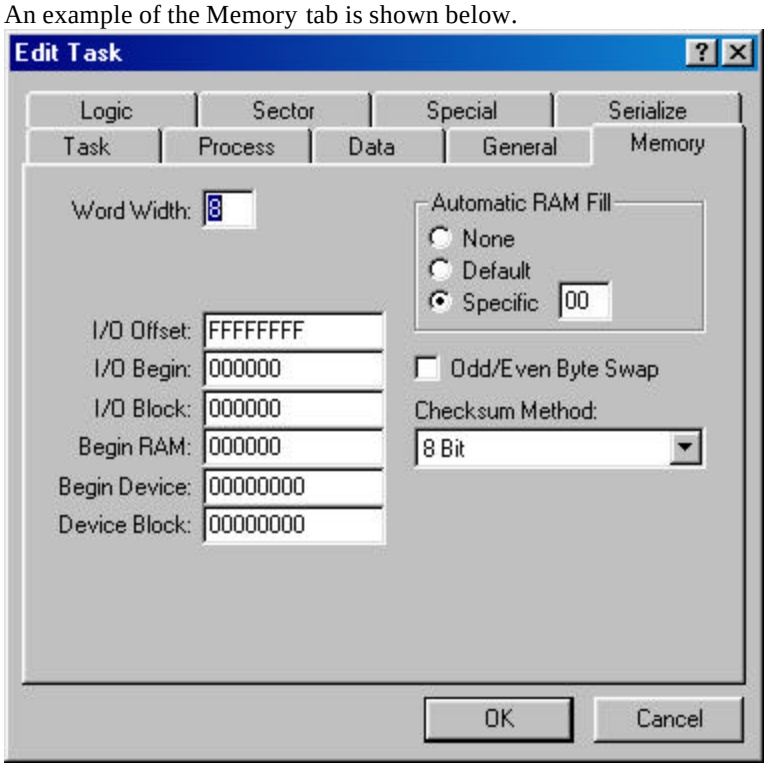


Memory

The Memory tab available from the Edit Task screen can be used to define the following data loading specifications.

- Word Width
- I/O Offset
- I/O Begin
- I/O Block
- Begin RAM
- Begin Device
- Device Block
- Automatic RAM Fill
- Odd/Even Byte Swap
- Checksum Method

An example of the Memory tab is shown below.

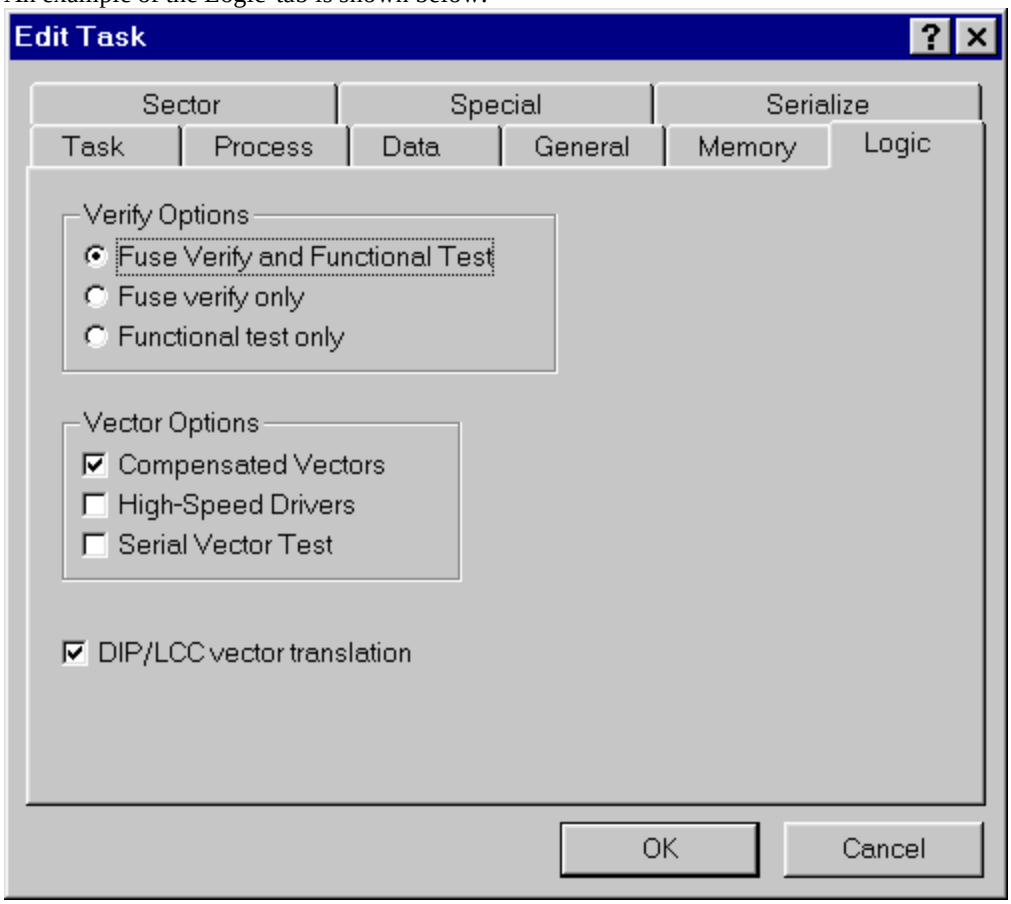


Logic

The Logic tab available from the Edit Task screen can be used to set various programming parameters for programmable logic devices.
Logic Parameters include:

- Verify Options - including Fuse Verify and Functional Test, Fuse verify only, and Functional test only
- Vector Options - Including compensated vectors, high speed drivers and serial vector test.
- DIP/LCC Vector Translation

The Logic tab settings have no effect when the target device is a memory or microprocessor device.
An example of the Logic tab is shown below.

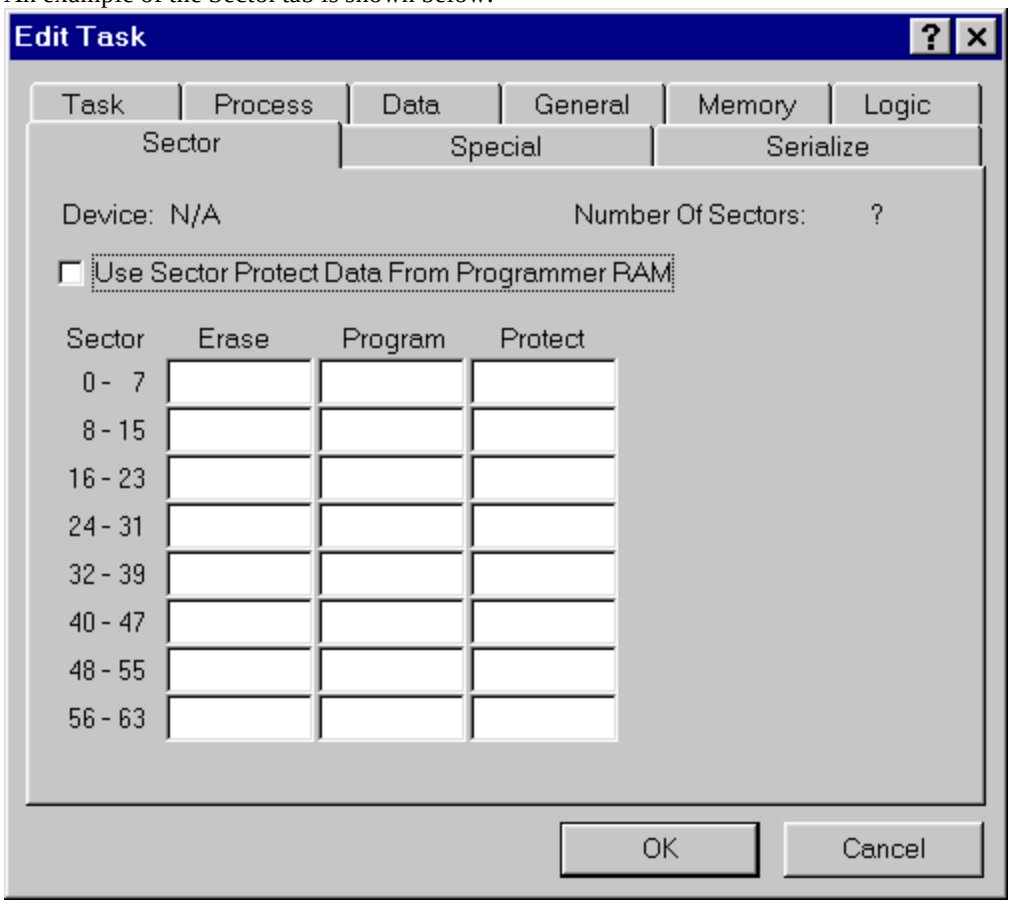


Sector

The Sector tab available from the Edit Task screen allows you to enable/disable the Erase, Program, and Secure capability of each device sector.
It also allows you to specify the sector data, allowing backwards compatibility with older data files that may contain sector protect data.
The options available on the Sector tab are listed below.

- Use Sector Data From Programmer RAM
- Erase Sector Data
- Program Sector Data
- Protect Sector Data

An example of the Sector tab is shown below.

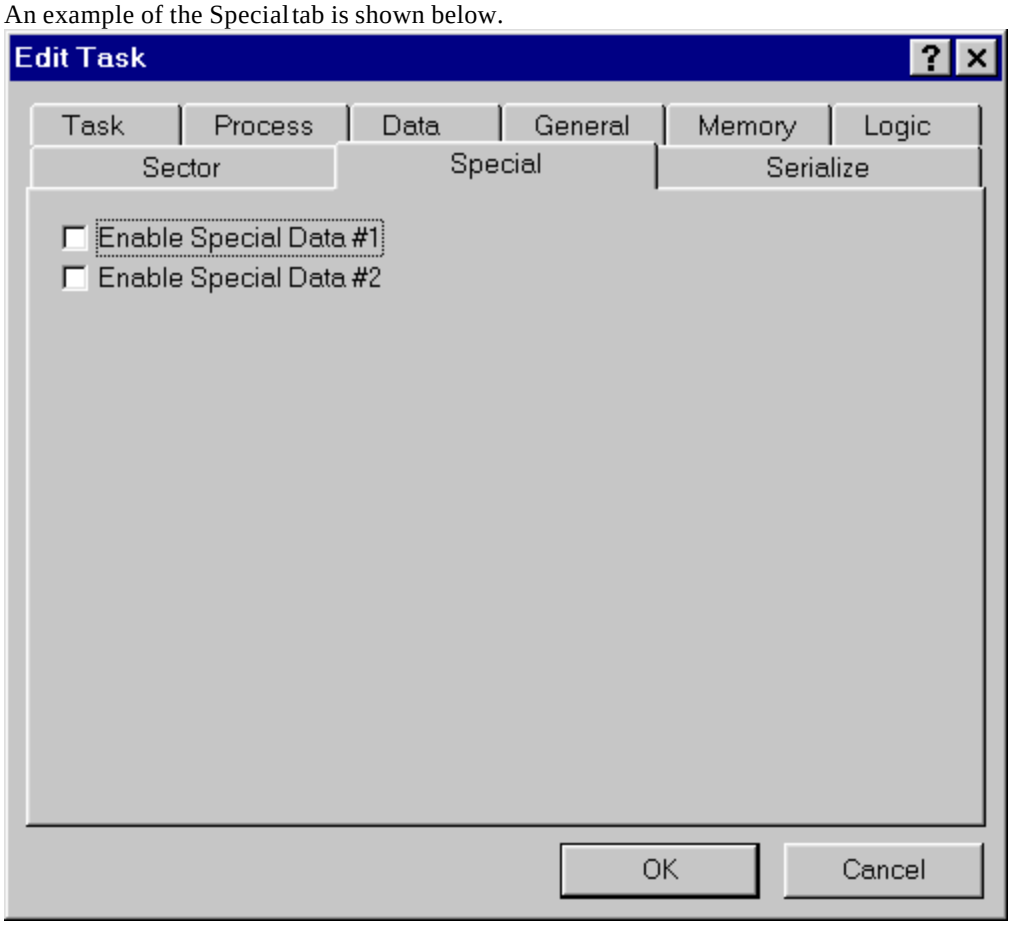


Special

The Special tab available from the Edit Task screen allows you to enable special data options to program selected locations in a device.

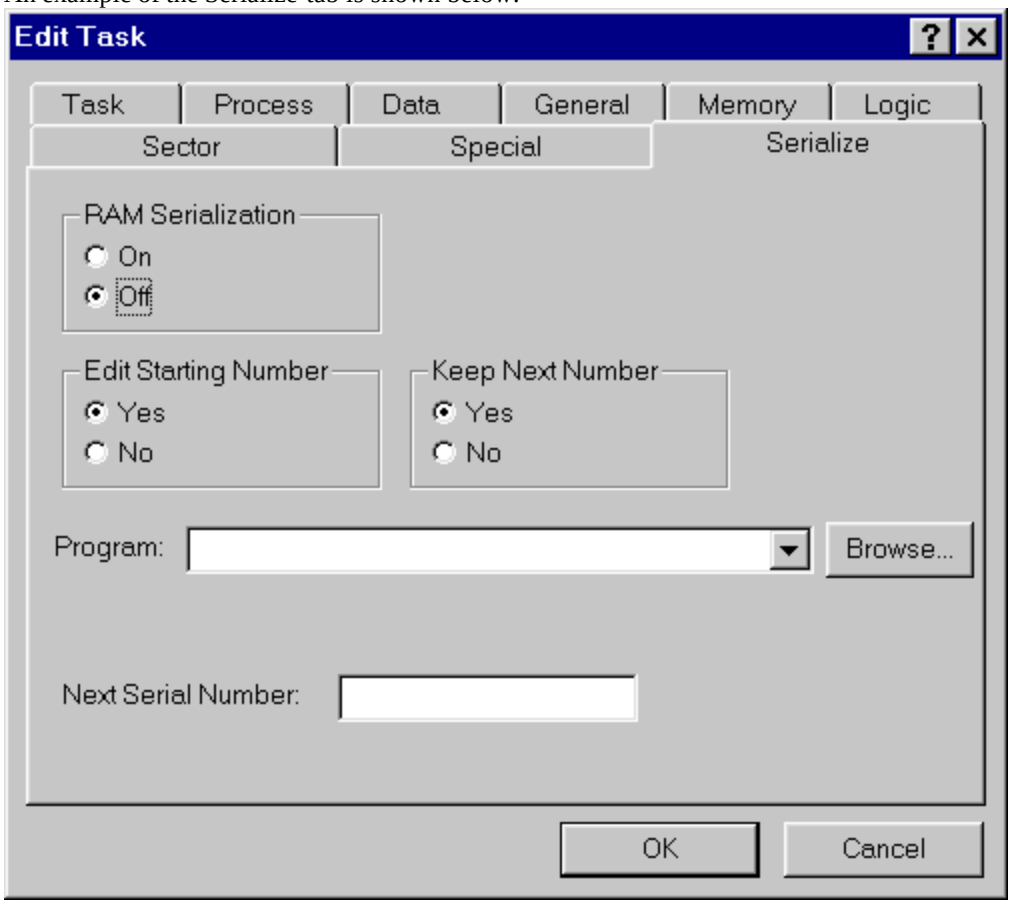
The Enable Special Data option is device-dependent. Refer to the Device List disk to determine if Enable Special Data applies to a specific device.

An example of the Special tab is shown below.



Serialize

The Serialize tab available from the Edit Task screen allows you to set serialization parameters for the task. Serialization is the generation of unique data for each device programmed.
An example of the Serialize tab is shown below.



TaskLink accomplishes serialization by running an external serialization program to generate the serial number for each device. The external serialization program (ESP) is run once at the beginning of the programming session and is run again each time a device is successfully programmed. The job of the ESP is to increment the serial number and to create the actual data that represents the serial number inside the part. This information is written to a disk file that TaskLink reads when the ESP is finished.
To do simple serialization of 8-bit devices involving a single binary (2- or 4-byte), ASCII decimal or ASCII hex number in each device, you can use the program SERIALIZ.EXE, which is supplied (with C source) with TaskLink.
For more complex serialization requirements, you can write your own external serialization program. All you need to know is:

- How to write a program that runs on a PC. Any language will do, provided the resultant program can accept command line arguments and output an ASCII file.
- The description of the command line arguments that TaskLink supplies to the ESP and the format of the disk file that the ESP creates for TaskLink.

See Also:
Help using SERIALIZ.EXE
Writing a Serialization Program

General

The General tab available from the Edit Task screen is used to set general programming parameters. These options apply to memory, microprocessor, and programmable logic devices.
General tab options include:

- Number of verify passes
- Custom Commands
- Reject Count
- Electronic I.D. Checking
- Bulk Erase of EE Devices
- Continuity Checking
- Security Option

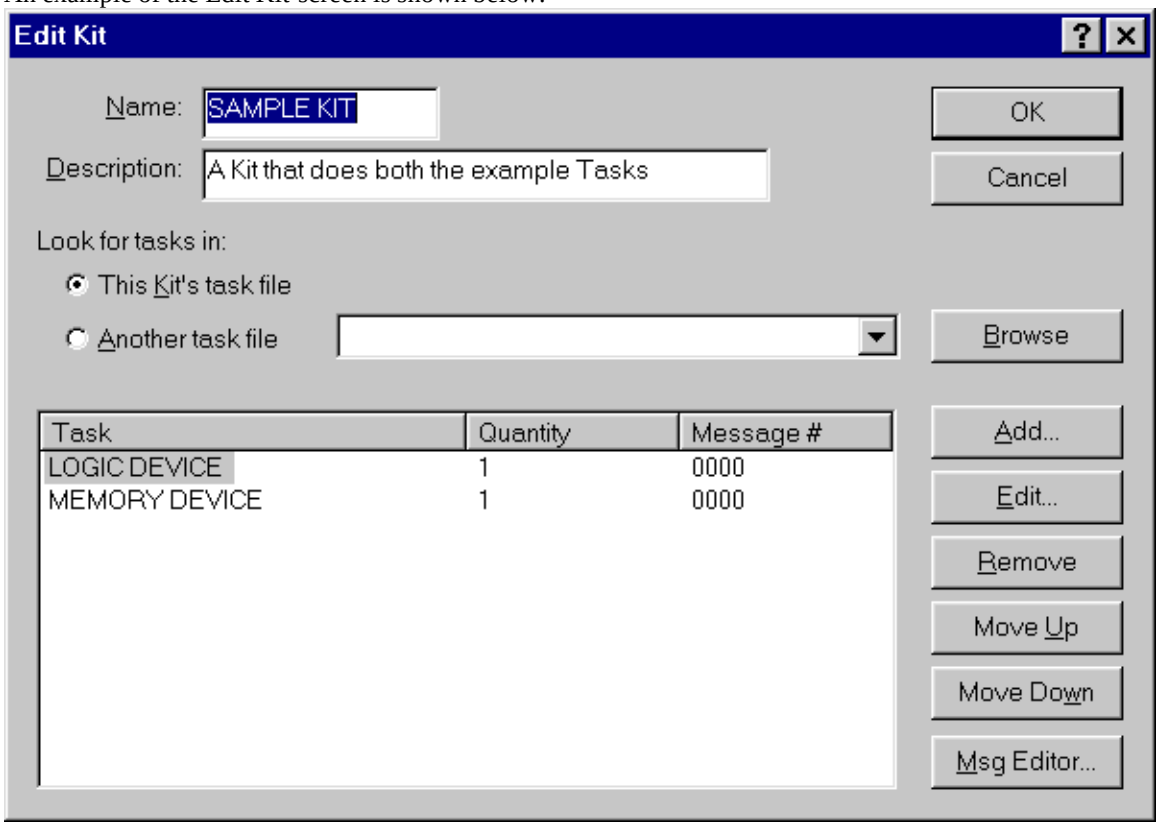
The General tab settings are always in effect, regardless of the device type selected.

Edit a Kit

The Edit Kit screen is available when you highlight a kit and click Edit from the Task/Kit Manager. Use the Edit Kit screen to change kit parameters. The Edit Kit screen includes:.

- Kit Name
- Kit Description
- Task file
- Task/Quantity/Message # list box

An example of the Edit Kit screen is shown below.

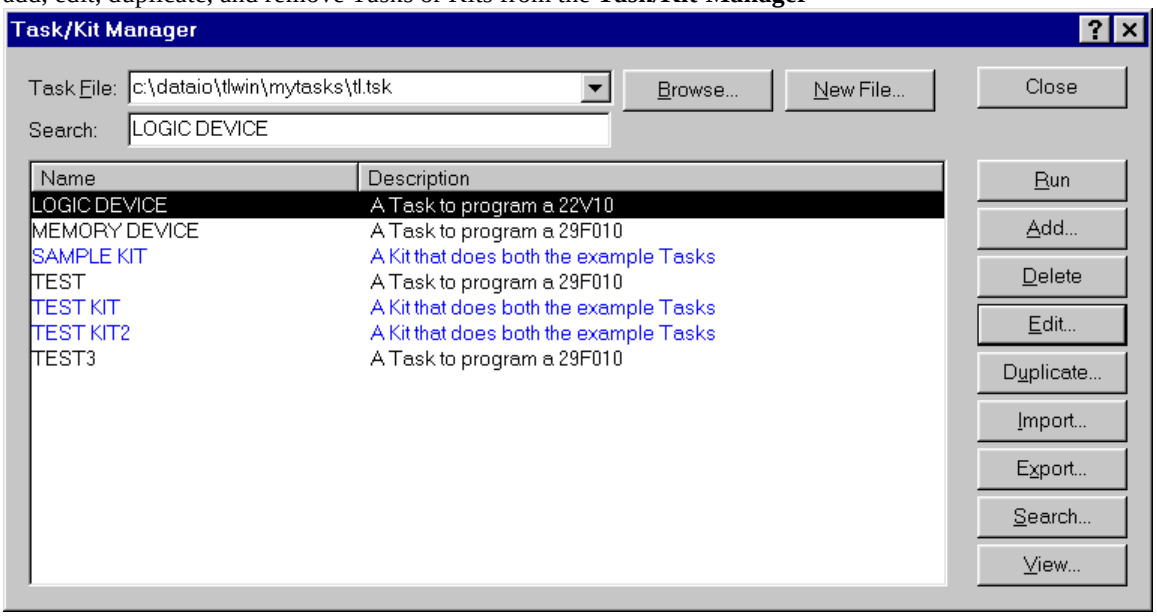


Kit parameters can be altered using the following options:
Browse - Searches other directories for .tsk Task Files.
Add - Adds new Tasks to the Kit
Edit - Changes existing Tasks.
Remove - Deletes Tasks from the Task/Quantity/Message # list.
Move Up - Reorders the Task sequence in a Kit so that the highlighted Task moves up in the sequence.

Move Down - Reorders the Task sequence in a Kit so that the highlighted Task moves down in the sequence.
Msg Editor - Creates notices to be delivered to operators prior to highlighted Tasks in the Task list.
When you exit the Edit Kit dialog box by clicking **OK**, your new selections are written to the specified Task/Kit Database file.

Use the Task/Kit Manager

For use only by administrators, the Task/Kit Manager is available by selecting **Task/Kit Manager** from the Task Menu. The Tasks and Kits available in the current Task file are listed in the list box. You can run, add, edit, duplicate, and remove Tasks or Kits from the **Task/Kit Manager**



Task File

At the Task Name box, enter the filename of a different Task file, click **Browse** to search other directories, or click **Import** to access other Task files in .tsk file format or ASCII text.

Run

Runs selected Tasks and Kits in TaskLink.

Add

Adds new Tasks and Kits to the Task/Kit Manager.

Delete

Deletes Tasks and Kits from Task/Kit Manager list box.

Edit

Opens either Edit Task or Edit Kit for changes to Tasks and Kits.

Duplicate

Copies highlighted task or kit data.

Import

Loads new Task files or ASCII text files from the network or PC.

Export

Saves Task files as ASCII text to other directories.

Search

Searches and Replaces PC Disk File, Device, and Programmer Type data in the Task file.

View

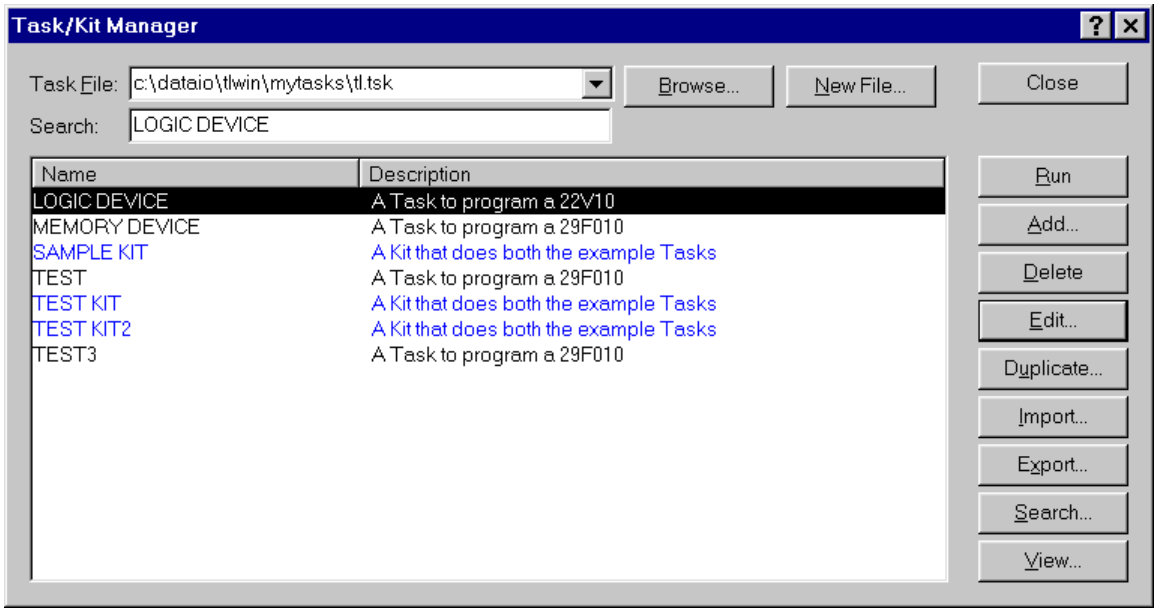
Opens a complete list of parameters for the highlighted Task/Kit.
See also:
What Are Tasks and Kits

Add a Task to a Kit

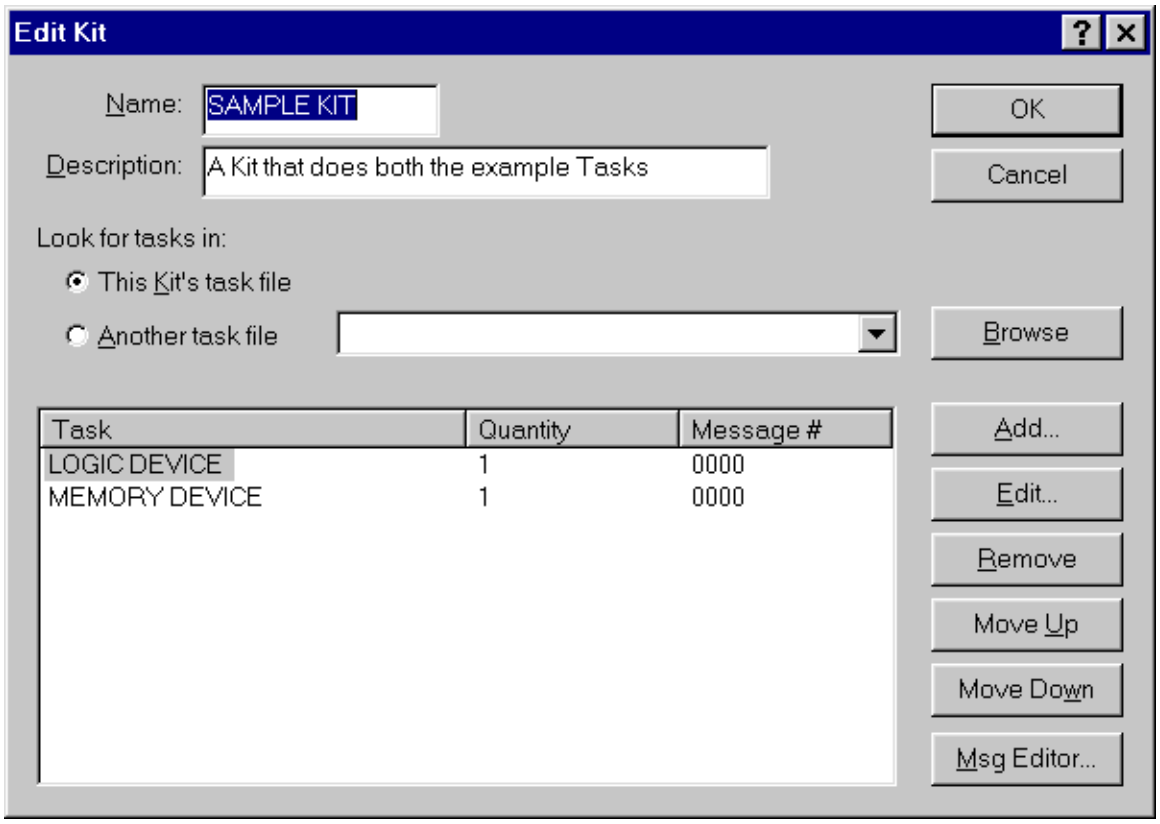
To add a Task to a Kit:

1. From the Task Menu, select Task/Kit Manager.

2. The Task/Kit Manager screen displays.



3. Highlight a Kit in the Task/Kit list box and click Edit.
4. The Edit Kit screen will appear.



5. Click Add and use the Task Name drop down box to highlight the Task from the Task list box.

***Note**: New Tasks cannot be created from Edit Kit. To create a new task, click Add at the Task/Kit Manager.*

6. Enter the Quantity Per Kit and Message Number, then click **OK.**
7. The Task you added is now part of the selected Kit.

Add New Users to TaskLink

To add a new user to TaskLink:

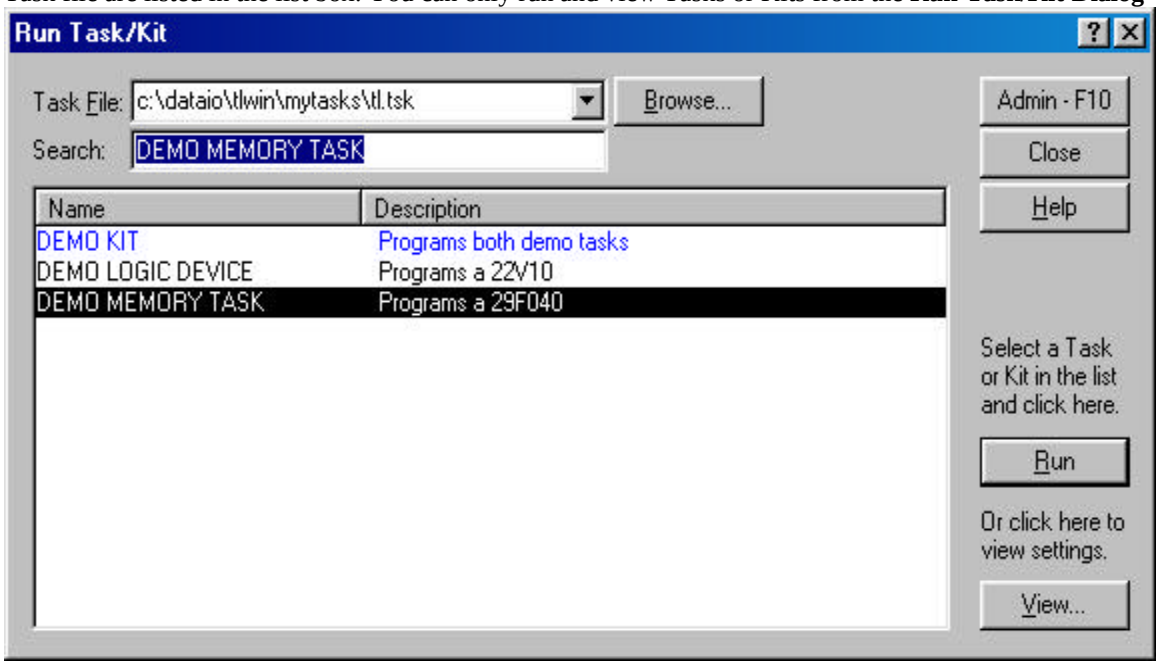
1. From the System Menu, select System Security.

2. Select Defined users with individual passwords and authorization levels and click Add User.
3. Enter the username, password and security level for each user.
4. Click OK.

***Note**: You must be in Administrator Mode to add new users to TaskLink.*

Run a Kit or Task

For use by operators, the Run Task/Kit dialog is available by selecting **Run** from the Task Menu or as the default dialog when TaskLink's security features are enabled. The Tasks and Kits available in the current Task file are listed in the list box. You can only run and view Tasks or Kits from the **Run Task/Kit Dialog**



Task File

At the Task File box, enter the filename of a different Task file, or click **Browse** to search other directories.

Run

Runs selected Tasks and Kits in TaskLink.

View

Opens a complete list of parameters for the highlighted Task/Kit.

Admin-F10

Uses your name and password to enter administrator mode. Only applied to operator mode when TaskLink security is enabled.

Search Capability

If you enter the name of a Task or Kit in the entry field, the highlight moves to the Task or Kit. If you enter the name of a non-existent Task or Kit in the entry field, the highlight moves to the position the Task or Kit would be if it existed. Use this feature to find a Task or Kit when you do not remember its name.

When you run a Task,

- The device type specified in the Task is selected in the programmer.
- Various programming/testing parameters are set as defined in the Task.
- The data file (if required) is sent to the programmer.
- TaskLink is configured to perform the set of operations specified in the Task.

TaskLink then enters the Process Devices screen, ready to perform the set of operations specified in the Task. The top line of the system status box shows the name of the currently loaded Task.

When you run a Kit, the above process is repeated for each Task in the Kit.

See also:

What Are Tasks and Kits

Use SERIALIZ.EXE

TaskLink accomplishes serialization by running an external serialization program to generate the serial number for each device. The external serialization program (ESP) is run once at the beginning of the programming session and again each time a device is successfully programmed.
The job of the ESP is to increment the serial number and to create the actual data that represents the serial number inside the part. This information is written to a disk file that TaskLink reads when the ESP is finished.
SERIALIZ.EXE is an external serialization program that is supplied with TaskLink. SERIALIZ.EXE can be used for two purposes:

- To generate simple serial numbers for programming into 8-bit memory devices (supported serial number formats are described below).
- As a template for generating your own custom external serialization program to create serial numbers in formats not supported by SERIALIZ.EXE. The source code SERIALIZ.C is included on the TaskLink distribution disk.

SERIALIZ.EXE can generate serial numbers with the following characteristics:

- Binary (1 to 4 bytes in length)
- ASCII decimal (1 to 10 bytes)
- ASCII hexadecimal (1 to 10 bytes)

Since SERIALIZ.EXE maintains the serial number internally as a 32-bit unsigned value, the maximum possible range of a serial number is 0 to 4,964,297,965 (0 to FFFFFFFF in hex). Choosing a field length of less than four in binary, ten in ASCII decimal, or eight in ASCII hex further limits the maximum possible serial number.
The amount added to the serial number each time can be any value from -32,768 to 32,767. For a typical application the value would be 1.
The bytes of the serial number (in any format) can be placed into the device in either forward or reverse order.
To use SERIALIZ.EXE, you must specify several items in TaskLink's Setup/Serialization dialog box.

1. Set the Serialization Method to File, set RAM Serialization to ON, and set the Edit Starting Number to Yes.
2. When creating a Task, you can set Keep Next Number to Yes to cause TaskLink to remember the next serial number to be used from one programming session to another.
3. Enter SERIALIZ.EXE in the Program entry field.
4. To select the serial number format and other options, enter command line arguments after the SERIALIZ.EXE program name.

The following is a complete list of command line parameters supported by SERIALIZ.EXE. As of version 1.70 TaskLink forces the serialization program command line to be lower case.
-a Address of serial number
Hexadecimal value indicating the device address of the serial number. For example, -a10 places the serial number at hexadecimal address 10 in the device. Default: Zero (-a0)
-c Compensation Address Hexadecimal value indicating the offset in the device at which the compensation value is to be located. For example, the command -c12345 would place the compensation value at hex address 12345.
This command causes the serialize program to generate a checksum compensation value for each serial number. In order for the compensation value to work correctly, the user data file or the master device must have the data at the two serial number addresses (serial number and compensation value) properly configured.
Set the data at one address to FF and the data at the other address to 00 so the sum of the two addresses is FF. When SERIALIZ.EXE creates a serial number, it also creates a value that when added to the serial number will add up to FF. Using this method causes the checksum of each device programmed to match the user data file.
For gang programmers only, this command must be used with the -m flag. If the -m flag is not supplied, the compensation address defaults to Disabled.
-d Delete Serial.dat file

Including the -d flag in the serialization parameters causes the serial.dat file to be deleted from the current directory. As of TaskLink version 1.60A, TaskLink does not automatically delete the serial.dat file. serializ.exe/serializ.c file version 1.42 or later is needed. Default: Omitted

-e Ethernet Mask

Hexadecimal value that indicates a fixed prefix value is to be placed in front of each generated serial number.

For example, -e0xABCD and a generated serial number of 1 would create the value ABCD1 to be programmed into the device.

This command must be used with the -z flag, which specifies the length of the mask value. Also the -l flag, which specifies the length of the serial number, must be set to the size of the total length. The length of the generated serial number is calculated by subtracting the total length from the mask length (gLen = tLen - eLen).

For example, serializ -l8 -z4 -e0xaa and a generated serial of 1 would create the value 00AA0001 to be programmed into the device.

-f Format of serial number

The following formats are supported.

-fb = Binary

-fd = ASCII Decimal

-fh = ASCII Hexadecimal

-fc = Binary Coded Decimal

Default: Binary (-fb)

-i Serial number increment value

Decimal value by which the serial number is incremented each time. The value may be negative. For example, -i2 increases the serial number by 2 for each device programmed. Default: 1 (-i1)

-l Length of serial number

Number of bytes the serial number occupies in the
device. The following field lengths are supported.

1 through 4 Binary

1 through 10ASCII

Default: 2 (-l2)

-m "Pure-Set" gang serialization mode flag

When you use a gang/set programmer, including this flag causes the serial numbers to be generated one device-size apart in programmer RAM instead of in a single block just beyond the fixed data in RAM. TaskLink then programs the devices in a single-pass operation with a set size equal to the number of sockets installed in the programmer. Without this flag, the programming would be done in two passes: the fixed data would be programmed in the initial gang programming pass (set size = 1) and the serial numbers would be programmed in a second pass (set size = number of sockets) with the block size set to the size of the serial information.

The two-pass method is considerably faster and does not require you to supply a duplicate image of the fixed data for each programmer socket. Omit this flag unless the device being programmed cannot be block programmed.

-o Byte order of serial number Serial numbers are placed in RAM with the most significant byte (MSB) or the least significant byte (LSB) at the first (lower) RAM address. This option applies to both binary and ASCII formats. The options are:

- -om = MSB at first RAM address (Motorola-style)
- -oi = LSB at first RAM address (Intel-style)
- Default: MSB at first RAM address (-om)
- -s Checksum calculation flag

Causes calculation of a new RAM checksum based on the sum of the serial number bytes and the initial checksum value (provided by TaskLink). Only meaningful if the initial RAM data (before serializing begins) contains zeroes in the serial number position(s). Used to provide a new checksum for display on the TaskLink screen as new serial numbers are generated. DOES NOT AFFECT DEVICE DATA.

-w Warning Limit

Can be used to set a warning level before the serial number limit is reached. The warning limit is used only when an ending serial number is specified. If the current serial number is within the ending serial number minus the warning limit, a warning message will be displayed on the TaskLink screen.

Default: Zero (-w0)
-z\g Ethernet Mask Length
Used to specify the length of the Ethernet Mask value.
See the -e flag for details.
When a processing session using SERIALIZ.EXE is begun, TaskLink prompts the operator for a starting serial number and an optional ending serial number. The starting number can be specified in either decimal or hex. To specify a hexadecimal number, begin the number with "0x," the standard C-language method of denoting hexadecimal. For example, the decimal number 16 could be entered as "0x10." SERIALIZ.EXE detects the presence of "0x" in the starting number and causes TaskLink to display any serial numbers on the screen in hexadecimal (with the "0x" prefix).
Example #1:\g To specify a 10-character ASCII hexadecimal serial number at address 10 (hexadecimal), increase the serial number by an increment of 2 for each device programmed, the Program entry field in the Setup/Serialization dialog box would look like this:
c:\\tasklink\\serializ.exe -l10 -fh -a10 -i2 -th
Example #2:\g To specify a 4-byte binary serial number at address 100 (hexadecimal) and a 4-byte compensation value at address 200 (hexadecimal) the Program entry field in the Setup/Serialization dialog box would look like this:
c:\\tasklink\\serializ.exe -l4 -fb -a100 -c200 -i1
Example #3:\g To specify a 4-byte binary serial number in hexadecimal display format that has a 2-byte prefix, for a total length of 6 bytes, the Program entry field in the Setup/Serialization dialog box would look like this:
c:\\tasklink\\serializ.exe -l6 -z2 -e0xAA -fb -a100 -i1

Write a Serialization Program

TaskLink accomplishes serialization by running an external serialization program to generate the serial number for each device. The external serialization program (ESP) is run once at the beginning of the programming session and again each time a device is successfully programmed. The job of the ESP is to increment the serial number and to create the actual data that represents the serial number inside the part. This information is written to a disk file that TaskLink reads when the ESP is finished.
An ESP called SERIALIZ.EXE is supplied with TaskLink and can be used to generate simple serial numbers in several different formats. To create serial number data in different formats than those provided by SERIALIZ.EXE, you can write your own ESP (perhaps by modifying the SERIALIZ.EXE source code). For example, you may want to generate network addresses to program into PROMs on Ethernet and other LAN cards.
To write an ESP, you need to know the following:

- How to write a program that runs on a PC. Any language will do provided the resulting program can accept command line arguments and output an ASCII file.
- Description of the command line arguments that TaskLink supplies to the ESP and the format of the disk file that the ESP creates for TaskLink.

The following is a detailed description of the information needed for the second item:

Communication with an ESP

TaskLink communicates with the ESP in two ways:

- The ESP receives data via arguments on the DOS command line used to invoke the ESP.
- The ESP returns data to TaskLink in the file SERIAL.DAT that the ESP creates in the current directory. This file is a line-oriented ASCII file containing the information needed to change the data in programmer RAM.

Command Line Arguments

TaskLink uses the following arguments on the command line to pass information to the ESP. These are always passed and do not need to be specified in the Program entry field in TaskLink's Setup/Serialization dialog box. Other command line arguments defined by the writer of the ESP may be entered in the Program entry field following the name of the ESP executable file.
-N serial_number
The ASCII representation of the current serial number (the number that the ESP is expected to format for programmer RAM). The format of serial_number is determined by the requirements of the ESP. In fact, if

the ESP is given the responsibility of keeping track of the next serial number to use, serial_number may not be present at all.

-E ending_serial_number

The ASCII representation of the last serial number that should be used during the programming session. This argument will not be present unless the operator has entered an ending serial number at the beginning of the session. The format of ending_serial_number is determined by the requirements of the ESP.

-D device_size

Hexadecimal size (in bytes) of the currently selected device. The ESP can use this information to generate a block of serial numbers for serializing devices in a gang/set programmer. The serial numbers must be placed in programmer RAM after the device data, so the ESP must know the size of the device.

-Q quantity of serial numbers needed

The number of serial numbers requested. When serialization is done on a gang/set programmer with multiple sockets, TaskLink will pass the ESP the number of sockets currently available on the programmer. The ESP can then generate multiple serial numbers, which are placed in a block of programmer RAM immediately following the device data (the size of the device is passed in via the -D parameter).

-S checksum

The checksum of programmer RAM at the beginning of the programming session (before any serial number information has been placed in programmer RAM). This is a hexadecimal value with as many digits as the programmer supports (up to eight). The ESP can use this checksum value to calculate a new RAM checksum when new serial number information is generated. Note that the checksum calculation can be accurate only if the RAM locations are set to zeros (or some value known to the ESP) before serialization begins. The ESP can return the new checksum to TaskLink, which will display it on the screen as devices are programmed. Use of this feature is optional and does not affect the serialization of the device.

-F

Flag that indicates this is the first call to the ESP at the beginning of the programming session. Use of this flag is determined by the ESP. An ESP might, upon seeing this flag, "check out" a beginning serial number from an external database.

-L

Flag that indicates this is the last call to the ESP after the programming session has been terminated. The last serial number that was produced by the ESP has NOT BEEN USED YET and is once again the value supplied to the ESP in serial_number. If the ESP is responsible for keeping the next serial number to be used, it can store the value in serial_number for use as the starting serial number in the next session.

For example, assuming a starting serial number of 12345, a 256-byte device and initial RAM that happens to have a zero checksum, the first call to SERIALIZ.EXE looks like this:

serializ.exe -F -D100 -S00000000 -N12345

The next call looks like this:

serializ.exe -D100 -S00000000 -N12346

Assuming the last serial number successfully programmed into
a device is 12500, the last call looks like this:

serializ.exe -L -D100 -S00000000 -N12501

The Output File SERIAL.DAT

The file SERIAL.DAT must be generated each time the ESP is called (except for the last time, indicated to the ESP by a -L on the command line). The file is a standard line-oriented ASCII file, each line being terminated by a carriage-return/line-feed sequence. Each line begins with a tag of the form Tnn, where nn is a decimal number. The tag identifies the type of data, if any, that the line contains. If the line contains additional data, the tag is followed by a colon (:) and then the data. The lines may appear in the file in any order, except for the T04 line, which marks the beginning of the serial number data and is by definition the last "T-record" in the file. A description of each type of T-record follows.

Mandatory records These T-records must always appear in the SERIAL.DAT file.

T01: CURRENT SERIAL NUMBER(S)

This line contains one or more ASCII representations of serial numbers that have been generated. If more than one serial number is generated, the numbers should be separated by commas.

Example:

T01:12345,12346,12347,12348,12349,1234A,1234B,1234C

T02: NEXT SERIAL NUMBER

This line contains the ASCII representation of the NEXT serial number to be used (after the number(s) that have just been formatted). This number will be the number that is passed to the ESP by TaskLink in the -N field of the command line the next time the ESP is called (unless you did not fill all the sockets on a multi-socket programmer, in which case TaskLink will pass back the next number that was not used). If the serialization is being done as part of a Task, TaskLink can store this number for you when the session is terminated for use as the starting serial number the next time the Task is run.
Example:
T02:1234D

T03: TRANSLATION FORMAT NUMBER

This line contains an ASCII representation of the Data I/O translation format number (1-99) for the serial number data in the file (which follows the T04 record). An ASCII translation format must be used, the records being terminated with a carriage-return/line-feed sequence. See your programmer manual for a description of supported formats.

***Note**: The C-language source for SERIALIZ.EXE, included with TaskLink, contains a routine for formatting a buffer of binary data into lines of Motorola format 95. This routine can be used for development of your own ESP.*

Example:
T03:95

T04: "BEGINNING OF SERIAL NUMBER DATA" MARKER

A line containing just the tag T04 marks the beginning of the serial number data. The contents of the SERIAL.DAT file from the line following the T04 record up to the end-of-file will be sent to the programmer verbatim. The data must be in one of the ASCII formats supported by the programmer being used. The format being used is specified in the T03 record. Use a Translation Format with address fields if you want the data to start anywhere except address 0 in the device.

Example SERIAL.DAT file

Here is an example file produced by an ESP. The ESP was called with the current serial number of 12345. The program formats the number as a long integer (four bytes) at address 0 in the device. The beginning- and end-of-file markers and comments are not part of the file.

<beginning-of-file>
T01:12345 ' Current serial number
T02:12346 ' Next serial number
T03:87' Format number for Motorola format 87
T04 ' Beginning of data marker
S2080000000803E98281 ' The formatted data in an S2 record
S9030000FC ' The format 87 end record
<end-of-file>

For more information on single-socket vs. gang serialization, see the topic "Single-socket vs. Gang Serialization."
Optional records
These T-records may be included in the file if needed.

T00: "PURE-SET" GANG SERIALIZATION FLAG (TASKLINK 1.1A)

Inclusion of this record tells TaskLink that the serial numbers generated are not "piled together" in a block just beyond the fixed data for the device in programmer RAM, but are placed in programmer RAM at the intended offset in the device and duplicated for each device to be programmed (up to a number of times equal to the number of sockets installed in the programmer) one device-size apart. TaskLink will then treat the programming operation as a one-pass operation with a set size equal to the number of installed sockets. Using the pure-set method causes slower programming and the need to duplicate the fixed data many times in programmer RAM (and thus the need for a large amount of RAM). The default two-pass method (one gang-programming pass for the fixed data and a second set-programming pass with the block size set to the size of the serial number data) is considerably faster and only requires one copy of the fixed data to be

loaded into programmer RAM. The two-pass method should be used unless the selected device cannot be block-programmed.

T05: FATAL-ERROR MESSAGE

If this record is included, TaskLink stops processing and displays the text string preceded by the message "Error message from external serialization program."

Example:

T05: Unable to generate serial number

T06: Limit-reached flag This record is included when the ESP has detected that the current serial number is greater than some limit (the last device programmed had the last allowable serial number). When this record is detected, TaskLink stops processing and displays "Ending serial number reached."

Example:

T06

T07: HEXADECIMAL ADDRESS OF SERIAL NUMBER IN DEVICE

This field is required only if you are serializing devices on a multi-socket gang programmer. It is a hexadecimal representation of the address in the device at which the serial number data is to be placed.

Example:

T07:00010

Be sure to read the Help topic "Single-socket vs. Gang Serialization."

T08: DECIMAL FIELD LENGTH OF SERIAL NUMBER IN DEVICE

This field is required only if you are serializing devices on a multi-socket gang programmer. It is a decimal representation of the number of bytes the serial number occupies in the device.

Example: T08:4

Be sure to read the Help topic "Single-socket vs. Gang Serialization."

T09: RAM CHECKSUM

This record contains the current RAM checksum. TaskLink displays this checksum in the System Status box while processing devices. This allows the external program to calculate the RAM checksum from the initial checksum value passed to the program on the command line. If this record is not included, TaskLink displays the initial checksum and does not change it during the programming session.

T11: WARNING MESSAGE

If this record is included, TaskLink stops processing and displays the text string. The operator will be given the choice to end the programming session or to continue.

Example: T11: Serial Number Warning Limit Reached

Example SERIAL.DAT file

Here is an example file produced by an ESP. The ESP was called with the current serial number of 12345. The program formats the number as a long integer (four bytes) at address 0 in the device. The beginning- and end-of-file markers and comments are not part of the file.

<beginning-of-file>
T01:12345 ' Current serial number
T02:12346 ' Next serial number
T03:87' Format number for Motorola format 87
T04 ' Beginning of data marker
S2080000000803E98281 ' The formatted data in an S2 record S9030000FC ' The format 87 end record
<end-of-file>

DATA I/O: SERVICE AND SUPPORT

Customer Support Offices

United States

Data I/O Customer Resource Center
Telephone: 800-3-DATAIO (800-332-8246)
Fax: 425-867-6972

Canada

Data I/O Canada
6725 Airport Road, Suite 102
Mississauga, Ontario
Canada, L4V1V2
Telephone: 905-678-0761
Fax: 905-678-7306

Japan

Data I/O Japan
Saisho Building 7F
8-1-14, Nishigotanda
Shinagawa-ku
Tokyo 141-0031
Telephone: 3-3779-2166
Fax: 3-3779-2203

Germany

Data I/O GmbH
Lochhamer Schlag 5
82166 Graefelfing
Germany
Telephone: 89-858-580
Fax: 89-858-5810

Other Countries

Contact your local Data I/O representative. For information on local representatives, go to www.dataio.com, and click on “Contact Us.”

End User Registration

If the end user for this product or your address has changed since the Registration Card was mailed, please notify Customer Support to ensure that you receive information about product enhancements. Be sure to include the product serial number, if available.

Software Update Service

A major benefit of Software Update Service is immediate support for new devices. In today's semiconductor industry, new devices are introduced to the market almost daily. Instead of waiting for the programmer update kit to receive new or updated device support, you can have this new support as soon as the algorithms are completed by Data I/O.

In addition to new device support, Software Update Service also gives you immediate access to modified or updated programming algorithms. These are new algorithms for existing devices that have been revised as a result of design or specification changes made by the device manufacturer. Most algorithm changes are critical to achieving optimum programming results, so it is extremely important to incorporate updated algorithms into your programming operations as soon as the modifications are made. Using an outdated algorithm can decrease your programming yields, increase your programming times, or affect the long-term reliability of your programmed devices.

UniFamily Programming System

The Keep Current Subscription Service is available for UniFamily Programming System:
UniSite Universal Programmer
3900/3980 Programming System
2900 Programming System
AutoSite Automated Production Programmer

Keep Current is a yearly subscription service that keeps your UniFamily programmer and documentation up-to-date with the latest device support and features. Keep Current gives you immediate access to new and modified programming algorithms via the Keep Current Data I/O web site. As a Keep Current subscriber, you can obtain new and updated device algorithms over the Keep Current web site before they are incorporated in a system software update.

Keep Current also provides periodic update kits that include all new and modified algorithms released since the last update, as well as new software and programmer features as they are introduced, along with User Manual updates.

Contact your local Data I/O Sales Representative for information about Keep Current Subscription Service for the UniFamily Programming System. To locate your Sales Representative, go to www.dataio.com and click on "Contact Us."

ProLINE-RoadRunner Programming System

Contact your local Data I/O Sales Representative for information about Software Update Service for the ProLINE Roadrunner Programming System. To locate your Sales Representative, go to www.dataio.com and click on "Contact Us."

Sprint Family Programming System

Contact your local Data I/O Sales Representative for information about Software Update Service for the Sprint Family Programming System. To locate your Sales Representative, go to www.dataio.com and click on "Contact Us."

Technical Assistance

Calling

To help us provide quick and accurate assistance, please be at your programmer or computer when you call, and have the following ready:

- Product version number
- Product serial number (if available)
- Detailed description of the problem you are experiencing
- Error messages (if any)
- Device manufacturer and part number (if device-related)
- Product manual

For information on how to contact the Customer Resource Center nearest you, go to www.dataio.comand click on “Contact Us.”

Electronic Mail

You can also reach Data I/O via e-mail. To help us provide quick and accurate assistance, please include the information listed above. Also, include your name, phone number, and e-mail address in your message, and send it to the following address:
techhelp@data-io.com

Data I/O Web Site

The Data I/O Home Page on the World Wide Web (WWW) includes links to online information about technical products, general information about Data I/O, a list of sales offices, and technical user information such as application notes and device lists.

To access the WWW, you will need an Internet account with Web access, and a Web browser. The address of the Data I/O Home Page is:
http://www.dataio.com

Warranty Information

Data I/O Corporation warrants this product against defects in materials and workmanship at the time of delivery and thereafter for a period of ninety (90) days.

The foregoing warranty and the manufacturers’ warranties, if any, are in lieu of all other warranties, expressed, implied or arising under law, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Data I/O maintains customer service offices throughout the world, each staffed with factory-trained technicians to provide prompt, quality service. For warranty service, see the Technical Support topic in the Help Index.

Command Line Parameters

The following command line parameters can be used with TaskLink:

Parameter	Description
tlwin	Runs TaskLink.
tlwin Qn	Program a fixed quantity of devices. Disables the prompt for pass quantity at run-time. Useful for batch mode operation.
tlwin “taskname”	TaskLink runs the specified Task and exits. The task must be in the current task file.
tlwin “task_database::task_name”	TaskLink runs the specified Task from the specified task database and exits.

Key Terms

Administrator Mode

Administrator mode is the TaskLink full-feature mode. Administrator mode allows access to all features of TaskLink, including those used for system setup and maintenance of Task/Kit databases.

Operator Mode

Operator mode is TaskLink's default mode when security is enabled. TaskLink starts within the Run Task/Kit dialog box. The operator selects a Task or Kit, and TaskLink sets up the system to process devices. After the required number of devices is processed, TaskLink returns to the Run Task/Kit dialog box.

Data Source

The source of the data currently in programmer RAM. The source may be the name of a disk file, "Master Device" or "Host Download." If the source of RAM data is unknown, the line contains a question mark (?).

Checksum

The checksum of the data in programmer RAM. The checksum is normally the checksum of RAM from the beginning of RAM to the end of the data that will fit into the selected device. If the Set Size is greater than one, the checksum is the sum of the checksums of each device in the set. If TaskLink has not yet calculated the RAM checksum this field contains a question mark (?).

Process

Combination of operations to be performed on the device.

Notes

Miscellaneous information about TaskLink status. The first line always shows whether or not all programming parameters are set to default values. If any programming parameters are set to non-default values, TaskLink will display each parameter in the list. Use the scrollbars to view the complete list of non-default parameters.

Counts

The count information is displayed only during device processing. The number of passed devices, the number of total devices, next starting serial number and the overall device yield are displayed. The number of failed parts is displayed in the job statistics section.

Times

The process time measurement runs from the moment the start button is pressed, until the process is complete and the "Waiting for Start" dialog is displayed.

%Yield

%Yield is the running percentage of devices that were programmed with no errors

Index

A

About Tasks and Kits, 8
Add a Task to a Kit, 59
Add New Users to TaskLink, 60

B

Blank Check Only, 36

C

Copy Data from Master Device to Programmer File, 31
Copy Data from PC File to Programmer File, 31
Copy Data from Programmer File to PC File, 31
Create Messages, 43
Custom Device File, 11
Customer Support Offices, 68

D

Data, 51
Data Menu, 27
Delete Programmer File, 33
Device Database Update, 40

E

Edit Kit, 57, 58
Edit Task, 47, 48
End User Registration, 68
Exit, 26

F

Features, 5
Fill Fuse Map, 32
Fill RAM, 32

G

General, 15, 57

H

Help Menu, 46
Host Command, 30

I

Illegal Bit Check Only, 36
Introduction, 4, 9

L

Load RAM from Host, 29
Load RAM from Master, 29
Load RAM from PC Network File, 27
Load RAM from Programmer File, 29
Log File, 25
Log File Format, 25
Logging, 45
Logic, 19, 53

M

Memory, 17, 52

O

Options, 43, 44

P

Password, 42
Port Settings, 45
Printer Setup, 42
Process, 13, 50
Process Devices, 35
Process Menu, 34
Program/Verify Only, 36
Programmer, 44
Programmer Interface, 40
Programmer Properties, 12

R

Restore Defaults, 24
Run, 61
Run Batch File, 38
Run Last Task/Kit (listed by filename), 39

S

Save RAM To PC/Network File, 30
Save RAM to Programmer File, 31
Sector, 54
Select Device, 10
Select Process, 36
Select Programming System, 41
Serialize, 23, 56
Session Data Logging, 24
Set Message Editor, 42
Setup Menu, 10
Simulation Mode, 43
Special, 22, 55
Subscription Service, 69
System Menu, 41
System Requirements, 5
System Security, 42

T

Task, 48, 49
Task Database Device Check, 38
Task Menu, 38
Task/Kit Manager, 58
TaskLink User Interface, 5
Technical Assistance, 70
Tools Menu, 40

U

UniSystem Programmers, 9
Username, 42
Using SERIALIZ.EXE, 64

V

Verify Only, 37
View/Edit RAM, 32
VT100 on Programmer Port, 40

W

Warranty Information, 70
Writing a Serialization Program, 64